NetModule Router NB3700

User Manual

Manual Version 1.0

NetModule AG, Switzerland

November 28, 2012

Contents

1 Welcome to NetModule **8**

2 Conformity **9**
2.1 Safety Instructions . . . 9
2.2 Declaration of Conformity . . . 10
2.3 Waste Disposal . . . 10
2.4 National Restrictions . . . 10
2.4.1 France . . . 10
2.4.2 Italy . . . 11
2.4.3 Latvia . . . 11
2.4.4 Luxembourg . . . 11
2.4.5 Norway . . . 11
2.4.6 Russian Federation . . . 11
2.4.7 Turkey . . . 12

3 Specifications **13**
3.1 Operating Elements . . . 13
3.2 Interfaces . . . 15
3.2.1 Overview . . . 15
3.2.2 USB 2.0 Host Port . . . 16
3.2.3 M12 Ethernet Connectors . . . 16
3.2.4 Power . . . 16
3.2.5 Digital Inputs and Outputs . . . 17
3.2.6 RS-232 Port (Optional) . . . 18

4 Installation **20**
4.1 Environmental Conditions . . . 20
4.2 Installation of the Router . . . 20
4.3 Installation of the SIM Card . . . 20
4.4 Installation of the WLAN Antennas . . . 21
4.5 Installation of the Local Area Network . . . 21
4.6 Installation of the Power Supply . . . 21

5 Configuration **22**
5.1 First Steps . . . 22
5.1.1 Initial Access . . . 22
5.1.2 Recovery . . . 23
5.2 HOME . . . 24
5.3 INTERFACES . . . 25
5.3.1 WAN . . . 25
5.3.2 Ethernet . . . 30
5.3.3 Mobile . . . 34
5.3.4 WLAN . . . 38
5.3.5 USB . . . 44
5.3.6 Serial Port . . . 46
5.3.7 Digital I/O . . . 48
5.3.8 GPS . . . 49
5.4 ROUTING . . . 51
5.4.1 Static Routes . . . 51
5.4.2 Extended Routing . . . 53
5.4.3 Bridging . . . 54
5.4.4 Mobile IP . . . 55
5.5 FIREWALL . . . 58
5.5.1 Administration . . . 58
5.5.2 Rules . . . 58
5.5.3 NAPT . . . 59
5.6 VPN . . . 62
5.6.1 OpenVPN . . . 62
5.6.2 IPsec . . . 67
5.6.3 PPTP . . . 72
5.6.4 Dial-In . . . 75
5.7 SERVICES . . . 76
5.7.1 SDK . . . 76
5.7.2 DHCP Server . . . 86
5.7.3 DNS Server . . . 88
5.7.4 DynDNS . . . 89
5.7.5 E-Mail . . . 90
5.7.6 Events . . . 91
5.7.7 SMS . . . 92
5.7.8 SSH/Telnet Server . . . 95
5.7.9 SNMP Agent . . . 97
5.7.10 Web Server . . . 101
5.7.11 Redundancy . . . 102
5.7.12 Voice Gateway . . . 104
5.8 SYSTEM . . . 107

5.8.1 System . . . 107
5.8.2 Authentication . . . 111
5.8.3 Software Update . . . 113
5.8.4 Configuration . . . 114
5.8.5 Troubleshooting . . . 117
5.8.6 Keys and Certificates . . . 120
5.8.7 Licensing . . . 123
5.9 LOGOUT . . . 124

6 Command Line Interface **125**
6.1 General Usage . . . 125
6.2 Print Help . . . 126
6.3 Getting Config Parameters . . . 127
6.4 Setting Config Parameters . . . 127
6.5 Getting Status Information . . . 128
6.6 Sending E-Mail or SMS . . . 128
6.7 Updating System Facilities . . . 129
6.8 Restarting Services . . . 129
6.9 Resetting System . . . 130
6.10 Rebooting System . . . 130
6.11 Running Shell Commands . . . 130
6.12 CLI-PHP . . . 131

7 Technical Support **136**

8 Legal Notice **137**

A Appendix **139**
A.1 Abbrevations . . . 139
A.2 System Events . . . 141
A.3 Factory Configuration . . . 143
A.4 SNMP VENDOR MIB . . . 144
A.5 SDK Examples . . . 151
A.6 SDK Sample SMS Control . . . 153

List of Figures

5.1 Home . . . 24

5.2 WAN Links . . . 26
5.3 WAN Settings . . . 27
5.4 Link Supervision . . . 28
5.5 LAN Ports . . . 30
5.6 LAN Link Settings . . . 31
5.7 LAN IP Configuration . . . 32
5.8 SIMs . . . 34
5.9 WWAN Interfaces . . . 36
5.10 WLAN Management . . . 38
5.11 WLAN Scan . . . 39
5.12 WLAN Interfaces . . . 40
5.13 WLAN Configuration . . . 41
5.14 WLAN IP Configuration . . . 42
5.15 USB Device Server . . . 45
5.16 Serial Port . . . 46
5.17 Static Routing . . . 51
5.18 Extended Routing . . . 53
5.19 Bridging . . . 54
5.20 Mobile IP . . . 56
5.21 NAPT Administration . . . 59
5.22 Inbound NAPT . . . 60
5.23 Outbound NAPT . . . 61
5.24 OpenVPN Administration . . . 62
5.25 OpenVPN Configuration . . . 64
5.26 OpenVPN Client Management . . . 66
5.27 IPsec Administration . . . 68
5.28 IPsec Configuration . . . 69
5.29 PPTP Administration . . . 72
5.30 PPTP Tunnel Configuration . . . 73
5.31 PPTP Client Management . . . 74
5.32 Dial-in Server Settings . . . 75
5.33 SDK Administration . . . 80
5.34 SDK Jobs . . . 82
5.35 SDK Testing . . . 84
5.36 DHCP Leases . . . 86
5.37 DHCP Server . . . 87
5.38 DNS Server . . . 88
5.39 Dynamic DNS Settings . . . 89
5.40 E-Mail Settings . . . 90
5.41 Event Notification Settings . . . 91
5.42 SMS Configuration . . . 93
5.43 SSH and Telnet Server . . . 95

5.44 SNMP Agent . . . 99
5.45 Web Server . . . 101
5.46 VRRP Configuration . . . 102
5.47 Voice Gateway . . . 104
5.48 Voice Client Configuration . . . 106
5.49 System . . . 107
5.50 Regional settings . . . 108
5.51 System information . . . 109
5.52 User Accounts . . . 111
5.53 Remote Authentication . . . 112
5.54 Manual File Configuration . . . 114
5.55 Automatic File Configuration . . . 115
5.56 Factory Configuration . . . 116
5.57 Log Viewer . . . 117
5.58 Tech Support File . . . 118
5.59 Keys and certificates management . . . 120
5.60 Licensing . . . 123
5.61 Logout . . . 124

List of Tables

3.1 NB3700 Status Indicators . . . 14
3.2 NB3700 Interfaces . . . 15
3.3 USB 2.0 Host Port Specification . . . 16
3.4 Ethernet Port Specification . . . 16
3.5 Pin Assignments of M12 Ethernet Connectors . . . 16
3.6 Power Connector Specifications . . . 17
3.7 Pin Assignments of Power Connector . . . 17
3.8 Isolated Digital Outputs Specification . . . 17
3.9 Isolated Digital Inputs Specification . . . 18
3.10 Pin Assignments of Digital Inputs and Outputs . . . 18
3.11 RS-232 Port Specification . . . 19
3.12 Pin Assignments of RS-232 Port . . . 19

4.1 Operating Conditions . . . 20

5.1 WAN Link Prerequisites . . . 25

5.2 Static Route Flags 52
5.5 SMS Control Commands 85
5.6 SMS Number Expressions 92
5.7 Certificate/Key Terms 121
5.8 Certificate Attributes 121

A.1 Abbreviations 140
A.2 System Events 142
A.3 SDK Examples 152

1. Welcome to NetModule

Thank you for purchasing a NetModule Router. This document should give you an introduction to the router and its features. The following chapters describe any aspects of commissioning the device, installation procedure and provide helpful information towards configuration and maintenance.

2. Conformity

This chapter provides general information for putting the router into operation.

2.1. Safety Instructions

NetModule routers must be used in compliance with any and all applicable national and international laws and with any special restrictions regulating the utilization of the communication module in prescribed applications and environments.

We would like to point out that only the original accessories, shipping with the router, must be used in order to prevent possible injury to health and damage to appliances and to ensure that all the relevant provisions have been complied with. Unauthorized modifications or utilization of unapproved accessories may void the warranty. The routers must not be opened. However, it is possible to replace any pluggable SIM cards even during operation.

All circuits connected to the interfaces of the router must comply with the requirements of Safety Extra Low Voltage (SELV) circuits and have to be designed for indoor use only. Interconnections must not leave the building nor penetrate the body shell of a vehicle. Possible antenna circuits must be limited to over-voltage transient levels below 1500 Volts according to IEC 60950-1, TNV-1 circuit levels using safety approved components.

NB3700 routers shall be only used with certified (CSA or equivalent) power supply, which must have a limited and SELV circuit output. They are basically designed for indoor use. Do not expose the communication module to extreme ambient conditions and protect the communication module against dust, moisture and high temperature.

We remind the user of the duty to observe the restrictions concerning the utilization of radio devices at petrol stations, in chemical facilities or in the course of blasting works in which explosives are used. Switch off the communication module when traveling by plane.

You need to pay heightened attention when using the communication module close to personal medical devices, such as cardiac pacemakers or hearing aids. NetModule routers may also cause interference in the nearer distance of TV sets, radio receivers and personal computers.

Avoid any installation of the antenna during a lightning. Always keep a distance of more than 40 cm from the antenna in order to reduce exposure to electromagnetic fields below the legal limits. This distance applies to $\frac{\lambda}{4}$- and $\frac{\lambda}{2}$-antennas. Larger distances may apply to antennas with higher gain.

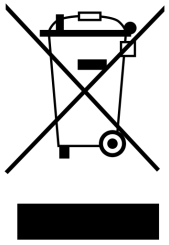
Any Ethernet cabling must be shielded, the Ethernet section of this manual provides

more information.
We highly recommended creating a copy of a working system configuration. It can be downloaded using the Web Manager and easily applied to a newer software release afterwards as we generally guarantee backward compatibility.

2.2. Declaration of Conformity

NetModule hereby declares that under our own responsibility that the routers comply with the relevant standards following the provisions of the *Council Directive 1999/5/EC*. The signed version of the *Declarations of Conformity* can be found at: http://www.netmodule.com/products

2.3. Waste Disposal



In accordance with the requirements of the *Council Directive 2002/96/EC* regarding Waste Electrical and Electronic Equipment (WEEE), you are urged to ensure that this product will be segregated from other waste at end-of-life and delivered to the WEEE collection system in your country for proper recycling.

2.4. National Restrictions

This product may be generally used in all EU countries (and other countries following the *EU directive 1999/5/EC*) without any limitation except for the countries mentioned below.

2.4.1. France

In case the product is used outdoors, the output power is restricted at some parts of the band. See the table below or check http://www.art-telecom.fr/ for more details.

Frequency	Power (EIRP)	Restrictions
2400-2454 MHz	100 mW (20 dBm)	Only for indoor applications
2454-2483.5 MHz	10 mW (10 dBm)	If used outdoors
5470-5725 MHz		Relevant provisions for the implementation of DFS mechanism described

2.4.2. Italy

This product meets the national radio interface regulations and requirements specified in the *National Frequency Allocation Table* for Italy. Unless operating within the boundaries of the owner's property, the use of this Wireless LAN product requires a general authorization. Please check http://www.comunicazioni.it for more details.

2.4.3. Latvia

The outdoor usage within the 2.4-GHz band requires authorization from the *Electronic Communications Office*. Please check http://www.esd.lv for more details.

2.4.4. Luxembourg

General authorization required for network and service apply.

2.4.5. Norway

Frequency	Restrictions
2400.0-2483.5 MHz	This band range cannot be operated in any geographical areas within a radius of 20km away from the center of Ny-Ålesund

2.4.6. Russian Federation

Frequency	Power (EIRP)	Restrictions
2400.0-2483.5 MHz	100 mW (20 dBm)	Only for indoor applications
5150-5250 MHz	100 mW (20 dBm)	Permitted to use only for indoor applications, closed industrial/warehouse areas and on board of aircrafts
5250-5350 MHz	100 mW (20 dBm)	1. Permitted to use for local networks of crew service communications on board of aircrafts in the area of the airport and at all stages of the flight. 2. Permitted to use for public wireless access local networks on board of a aircraft during the flight but at a altitude of not less than 3000 m

Frequency	Power (EIRP)	Restrictions
5650-5825 MHz	100 mW (20 dBm)	Permitted to use on board of the aircraft during a flight at a altitude not less than 3000 m

2.4.7. Turkey

Frequency	Restrictions
5470-5725 MHz	Not implemented

3. Specifications

There is currently one model of NB3700 available:

- NB3700 Mobile & WLAN

Due to its modular approach, the NB3700 router and its hardware components can be arbitrarily assembled according to its indented usage or application. Please contact us in case of special project requirements.

3.1. Operating Elements

The following table describes the NB3700 status indicators. The color of the LED represents the signal quality for wireless links.

- red means low
- yellow means moderate
- green means good or excellent

Label	Color	State	Function
Status	●	blinking	The device is busy due to startup, software or configuration update.
	●	on	The device is ready. The captions of the top bank apply.
	●	on	The device is ready. The captions of the bottom bank apply.
Mob1	●●●	on	Mobile connection 1 is up.
	●	blinking	Mobile connection 1 is being established.
	❍	off	Mobile connection 1 is down.
Mob2	●●●	on	Mobile connection 2 is up.
	●	blinking	Mobile connection 2 is being established.
	❍	off	Mobile connection 2 is down.
VPN	●	on	VPN connection is up.
	❍	off	VPN connection is down.
WLAN	●●●	on	WLAN connection is up.

Label	Color	State	Function
	●	blinking	WLAN connection is being established.
	❍	off	WLAN connection is down.
GPS	●	on	GPS is turned on and a valid NMEA stream is available.
	❍	off	GPS is turned off or no valid NMEA stream is available.
Voice	●	on	A voice call is currently active.
	❍	off	No voice call is active.
DO1	●	on	Normally open output port 1 is closed.
	❍	off	Normally open output port 1 is open.
DO2	●	on	Normally closed output port 2 is closed.
	❍	off	Normally closed output port 2 is open.
DI1	●	on	Input port 1 is set.
	❍	off	Input port 1 is not set.
DI2	●	on	Input port 2 is set.
	❍	off	Input port 2 is not set.

Table 3.1.: NB3700 Status Indicators

3.2. Interfaces

3.2.1. Overview

Label	Panel	Function
SIM 1	Front	SIM 1, it can be assigned dynamically to any modem by configuration.
SIM 2	Front	SIM 2, it can be assigned dynamically to any modem by configuration.
USB	Front	USB 2.0 host port, can be used as USB device server or for software/configuration updates.
Ethernet 1-4	Front	Ethernet switch ports, can be used as LAN or WAN interface.
Ethernet 5	Front	Additional Ethernet switch port (represents an RS232 port for NB3700-4ES)
⏚	Front	Earth protection connector (earthing is optional), connected to the ground of the power supply V_{GND}. If used, connect a yellow-green marked cable with at least $6mm^2$ copper area. Avoid corrosion and protect the screws against loosening
Power	Front	Power supply (galvanically isolated)
Digital I/O	Front	Galvanically isolated digital I/O M12 connector
Mob 1	Front	TNC female connector for mobile antenna 1
WLAN 1	Front	TNC female connector for first WLAN antenna (main)
GPS	Front	TNC female connector for GPS antenna
Mob 2	Front	TNC female connector for mobile antenna 2
WLAN 2	Front	TNC female connector for second WLAN antenna (diversity)
Reset	Front	Reset button, press at least 3 seconds for reboot and at least 5 second for a factory reset. The start of the factory reset is confirmed by all LEDs lighting up for a second. The button can be released then again.

Table 3.2.: NB3700 Interfaces

3.2.2. USB 2.0 Host Port

The USB 2.0 host port has the following specification:

Feature	Specification
Speed	Low, Full & Hi-Speed
Current	max. 500 mA

Table 3.3.: USB 2.0 Host Port Specification

3.2.3. M12 Ethernet Connectors

Specification

The five Ethernet ports have the following specification:

Feature	Specification
Isolation	1500 Vrms
Speed	10/100 Mbps
Mode	Half- & Full-Duplex
Crossover	Automatic MDI/MDI-X
Connector type	M12, 4 poles, D-coded female

Table 3.4.: Ethernet Port Specification

Pin Assignment

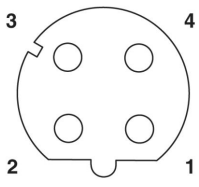
Pin	Signal	Pinning
1	Tx+	
2	Rx+	
3	Tx−	
4	Rx−	

Table 3.5.: Pin Assignments of M12 Ethernet Connectors

3.2.4. Power

The power connector has the following specifications:

Feature	Specification
Power supply nominal voltages	24 V_{DC}, 36 V_{DC} and 48 V_{DC} according to EN 50155
Voltage range	12 V_{DC} to 60 V_{DC} (−15% / +5%)
Max. power consumption	15 W
DC isolation	yes
Power Interruption Class S2	Sustains interruptions up to 10 ms, there are no batteries included
Connector type	M12, 4 poles, A-coded male

Table 3.6.: Power Connector Specifications

Pin Assignment

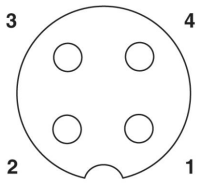
Pin	Signal	Pinning
1	V+ (12−60 V_{DC})	
2	Not connected	
3	V_{GND}	
4	Not connected	

Table 3.7.: Pin Assignments of Power Connector

3.2.5. Digital Inputs and Outputs

Isolated Outputs

The isolated digital output ports have the following specification:

Feature	Specification
Number of output ports	2
Limiting continuous current	1 A
Maximum switching voltage	60 V_{DC}, 42 V_{AC} (V_{rms})
Maximum switching capacity	60 W

Table 3.8.: Isolated Digital Outputs Specification

Isolated Inputs

The isolated digital input ports have the following specification:

Feature	Specification
Number of inputs	2
maximum input voltage	40 V_{DC}
Minimum voltage for level 1 (set)	7.2 V_{DC}
Maximum voltage for level 0 (not set)	5.0 V_{DC}

Table 3.9.: Isolated Digital Inputs Specification

Note: A negative input voltage is not recognized.

Pin Assignment

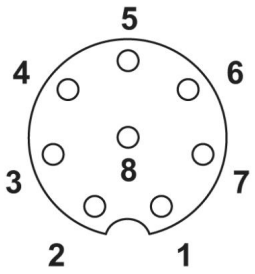
Pin	Signal	Pinning
1	DI1+	
2	DI1−	
3	DI2+	
4	DI2−	
5	DO1: Dry contact relay normally open	
6	DO1: Dry contact relay normally open	
7	DO2: Dry contact relay normally closed	
8	DO2: Dry contact relay normally closed	

Table 3.10.: Pin Assignments of Digital Inputs and Outputs

3.2.6. RS-232 Port (Optional)

The RS-232 port (if present) has the following specification:

Feature	Specification
Protocol	3-wire RS-232 (TXD, RXD, GND)

Feature	Specification
Baud rate	300, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600, 115 200
Data bits	7 bit, 8 bit
Parity	none, odd, even
Stop bits	1, 2
Software flow control	None, XON/XOFF
Hardware flow control	None

Table 3.11.: RS-232 Port Specification

Pin Assignment

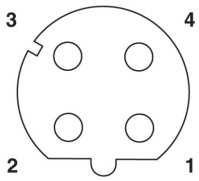
Pin	Signal	Pinning
1	GND	
2	RxD	
3	not connected	
4	TxD	

Table 3.12.: Pin Assignments of RS-232 Port

4. Installation

4.1. Environmental Conditions

The following precautions must be taken before installing a NB3700 router:

- Avoid direct solar radiation
- Protect the device from humidity, steam and aggressive fluids
- Guarantee sufficient circulation of air around the device
- The device is for indoor use only

Parameter	Rating
Input Voltage	12 V_{DC} to 48 V_{DC} (−15% / +20%)
Operating Temperature Range	−25 °C to +70 °C
Humidity	0 to 95% (non-condensing)
Altitude	up to 4000m
Over-Voltage Category	II
Pollution Degree	2
Ingress Protection Rating	IP40 (with SIM and USB covers mounted)

Table 4.1.: Operating Conditions

4.2. Installation of the Router

NB3700 is designed for mounting it on a worktop or wall. Please consider the safety instructions and the environmental conditions in chapter 2.

4.3. Installation of the SIM Card

SIM cards can be inserted by sliding it into one of the designated holes on the front panel. By using a small paper clip (or similar) you will need to press it a bit until it snaps into place. For removing the SIM, you will need to push it again in the same manner. The SIM card will then rebounce and can be pulled out.

SIMs can be assigned flexibly to any modem in the system. It is also possible to switch a SIM to a different modem during operation, for instance if you want to use another

provider upon a certain condition. However, a SIM switch usually takes about 10-20 seconds which can be bypassed (e.g. at bootup) if SIMs are installed reasonably. Using only a single SIM with one modem, it should be preferably placed into the SIM 1 holder. For systems which should operate two modems with two SIMs in parallel, we recommend to assign **Mobile 1** to SIM 1 and **Mobile 2** to SIM 2.
Further information about SIM configuration can be found in chapter 5.3.3.

4.4. Installation of the WLAN Antennas

Any WLAN antennas must be mounted to the connectors **WLAN1** and **WLAN2**. The number of attached antennas can be configured in the software. If only one antenna is used, it must be attached to **WLAN1**. However, for better diversity and thus better throughput and coverage, we highly recommend using two antennas.

4.5. Installation of the Local Area Network

Up to two 10/100 Mbps Ethernet devices can be directly connected to the router, further devices can be attached via an addtional Ethernet switch. Please ensure that the connector has been plugged in properly and remains in a fixed state, you might otherwise experience sporadical link loss during operation. The Link/Act LED will lit up as soon as the device has synced. If not, it might be necessary to configure a different link setting as described in chapter 5.3.2.

4.6. Installation of the Power Supply

The router can be powered with an external source supplying between 12 V_{DC} and 48 V_{DC}. It is to be used with a certified (CE or equivalent) power supply, which must have a limited and SELV circuit output. The router is now ready for getting engaged.

5. Configuration

The following chapters give information about setting up the router and configuring its features as provided with system software 3.6.

5.1. First Steps

NetModule routers can be easily set up by using the HTTP-based configuration interface, called the Web Manager. Advanced users may also use the Command Line Interface (CLI) and set configuration parameters directly.

You may also upload configuration files via SNMP, SSH, HTTP or USB in case you intend to deploy a larger numbers of routers.

The Web Manager is supported by the latest web browsers (e.g. Microsoft Internet Explorer 9, Mozilla Firefox 15.0 and many others). Please ensure to have JavaScript turned on. The IP address of Ethernet1 is `192.168.1.1` and the Dynamic Host Configuration Protocol (DHCP) is activated on the interface by default. The following steps need to be taken to establish your first Web Manager session:

1. Connect the Ethernet port of your computer to the Ethernet1 port of the router using a standard CAT5 cable with RJ45 (or M12) connectors.
2. If not yet activated, enable DHCP on your computer's Ethernet interface so that an IP address can be obtained automatically from the router. This usually takes a short amount of time until your PC has received the corresponding parameters (IP address, subnet mask, default gateway, name server). You may track the progress by having a look to your network control panel and check whether your PC has correctly retrieved an IP address of the range `192.168.1.100` to `192.168.1.199`.
3. Launch your favorite web browser and point it to the IP address of the router (the URL is http://192.168.1.1).
4. Please follow the instructions of the Web Manager for configuring the router. Most of the menus are self-explanatory, further details are given in the following chapters.

5.1.1. Initial Access

In factory state you will be prompted for a new administrator password. Please choose a password which is both, easy to remember but also robust against dictionary attacks (such as one that contains numbers, letters and punctuation characters). The password shall have a minimum length of 6 characters. It shall contain a minimum of 2 numbers and 2 letters.

Please note that the admin password will be also applied for the root user which can be used to access the device via the serial console, telnet, SSH or to enter the bootloader. You may also configure additional users which will only be granted to access the summary page or retrieve status information but not to set any configuration parameters.

A set of services (USB Autorun, CLI-PHP) are by default activated in factory state and will be disabled as soon as the admin password has been set. They can be enabled again afterwards in the relevant sections.

5.1.2. Recovery

Following actions might be taken in case the router has been misconfigured and cannot be reached anymore:

1. Factory Reset: You can initiate a reset back to factory settings via the Web Manager, by running the command `factory-reset` or by pressing the reset button. The latter would require a slim needle or paper clip which must be inserted into the hole below the USB port. The button must be hold pressed for up to 5 seconds until all LEDs flash up.

5.2. HOME

This page provides a status overview of established connections and enabled features.

It offers a summary about the administrative and operational status of the router's interfaces. Further details about any enabled Wide Area Network (WAN) links (such as the IP addresses, network information, signal strength, etc.) will be shown on additional tabs. The information about the amount of downloaded/uploaded data is stored in non-volatile memory, thus survive a reboot of the system.

The counters can be reset by pressing the *Reset* button.

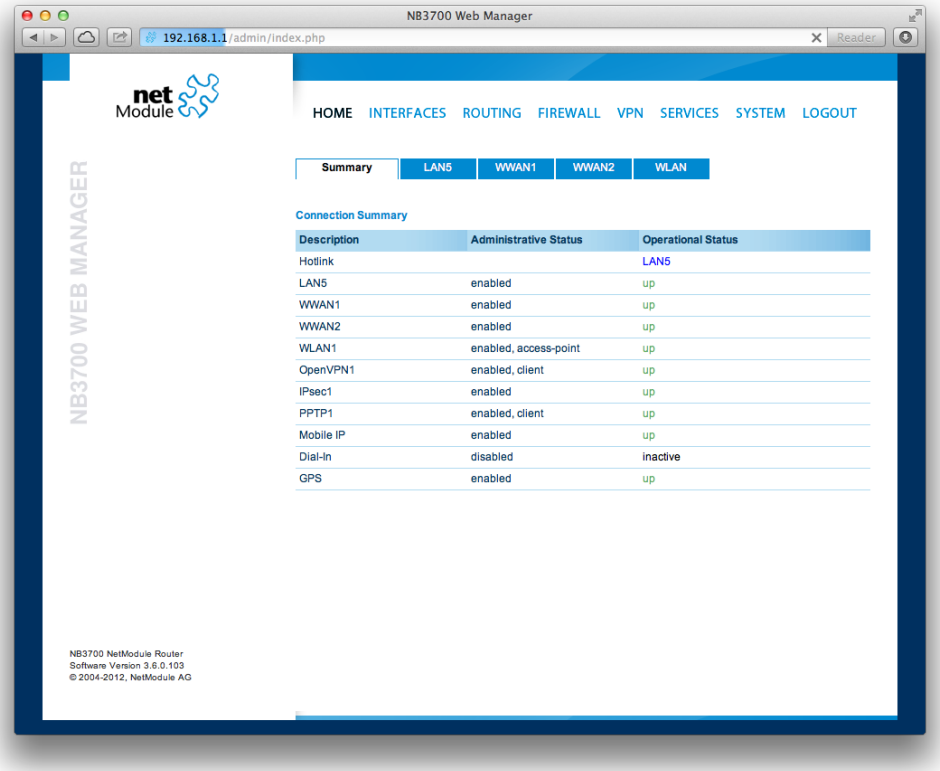

NB3700 Web Manager
192.168.1.1/admin/index.php
Reader
net Module
HOME INTERFACES ROUTING FIREWALL VPN SERVICES SYSTEM LOGOUT
NB3700 WEB MANAGER
Summary LAN5 WWAN1 WWAN2 WLAN
Connection Summary
Description | Administrative Status | Operational Status
Hotlink | | LAN5
LAN5 | enabled | up
WWAN1 | enabled | up
WWAN2 | enabled | up
WLAN1 | enabled, access-point | up
OpenVPN1 | enabled, client | up
IPsec1 | enabled | up
PPTP1 | enabled, client | up
Mobile IP | enabled | up
Dial-In | disabled | inactive
GPS | enabled | up
NB3700 NetModule Router
Software Version 3.6.0.103
© 2004-2012, NetModule AG

Figure 5.1.: Home

5.3. INTERFACES

5.3.1. WAN

Link Management

Depending on your hardware model, WAN links can be made up of either Wireless Wide Area Network (WWAN), Wireless LAN (WLAN), Ethernet or PPP over Ethernet (PPPoE) connections. Please note that each WAN link has to be configured and enabled in order to appear on this page.

Generally, a link will be only dialed or declared as up if the following prerequisites are met:

Condition	WWAN	WLAN	ETH	PPPoE
Modem is registered	X			
Registered with valid service type	X			
Valid SIM state	X			
Sufficient signal strength	X	X		
Client is associated		X		
Client is authenticated		X		
Valid DHCP address retrieved	X	X	X	X
Link is up and holds address	X	X	X	X
Ping check succeeded	X	X	X	X

Table 5.1.: WAN Link Prerequisites

The menu can be used further to prioritize your WAN links. The highest priority link which has been established successfully will become the so-called **hotlink** which holds the default route for outgoing packets.

In case a link goes down, the system will automatically switch over to the next link in the priority list. You can configure each link to be either established when the switch occurs or permanently in order to minimize link downtime.

1st priority: The primary link which will be used whenever possible.

2nd priority: The first fallback link, it can be enabled permanently or being dialed as soon as Link 1 goes down.

3rd priority: The second fallback link, it can be enabled permanently or being dialed as soon as Link 2 goes down.

4th priority: The third fallback link, it can be enabled permanently or being dialed as

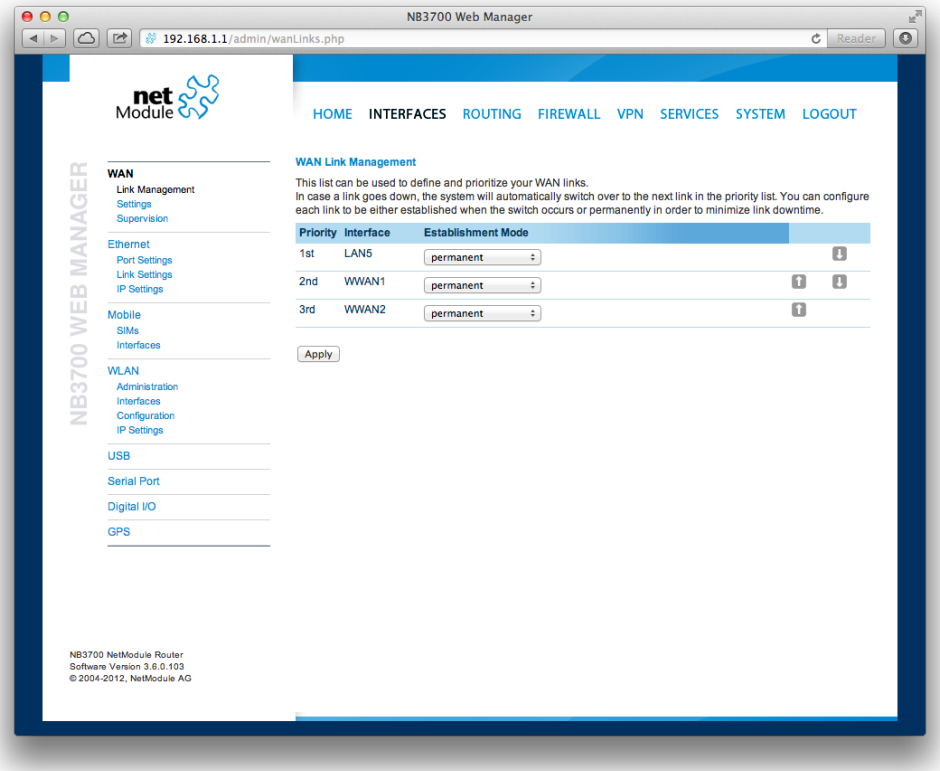


Figure 5.2.: WAN Links

soon as Link 3 goes down.

Links are being triggered periodically and put to sleep in case it was not possible to establish them within a certain amount of time. Hence it might happen that permanent links will be dialed in background and, as soon as they got established, replace low-priority links again.

We recommend to use the `permanent` option for WAN links in general. However, in case of time-limited mobile tariffs for instance, the `switchover` option might be applicable.

Settings

This page can be used to configure WAN specific settings like the Maximum Segment Size (MSS). The MSS corresponds to the largest amount of data (in bytes) that the router can handle in a single, unfragmented TCP segment. In order to avoid any negative side effects the number of bytes in the data segment and the headers must not add up to more than the number of bytes in the Maximum Transmission Unit (MTU). The MTU can be configured per each interface and corresponds to the largest packet size that can be transmitted.

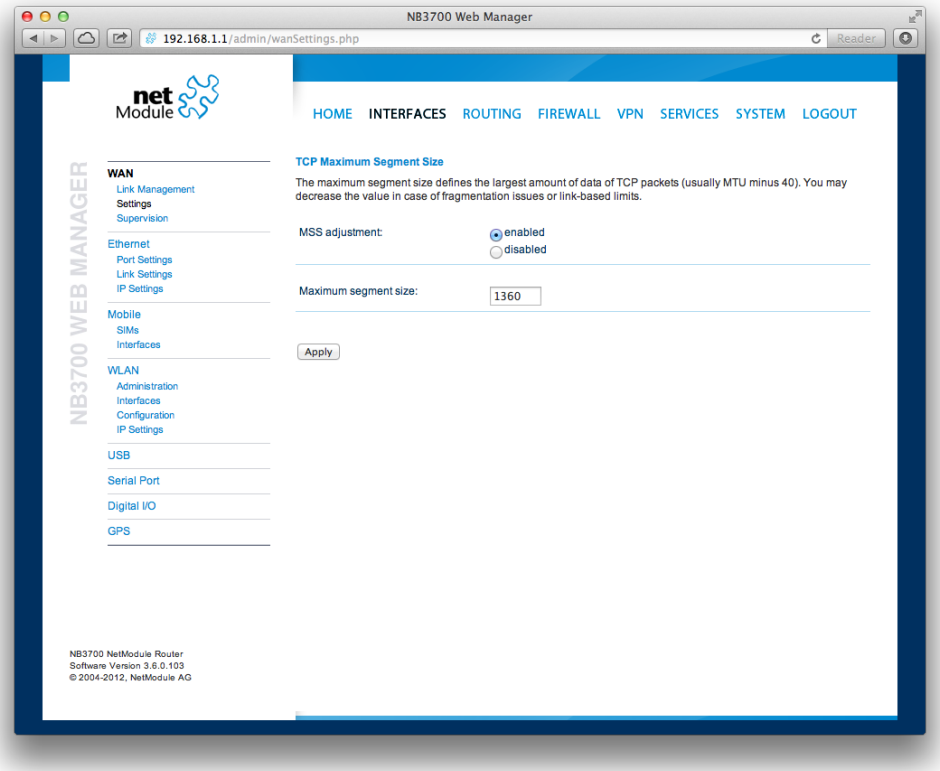


Figure 5.3.: WAN Settings

MSS adjustment: Enable or disable MSS adjustment on WAN interfaces.

Maximum segment size: Maximum number of bytes in a TCP data segment.

Supervision

Network outage detection can be performed by sending pings on each link to some authoritative hosts. A link will be declared as down in case all trials have failed and only as up if at least one host can be reached.

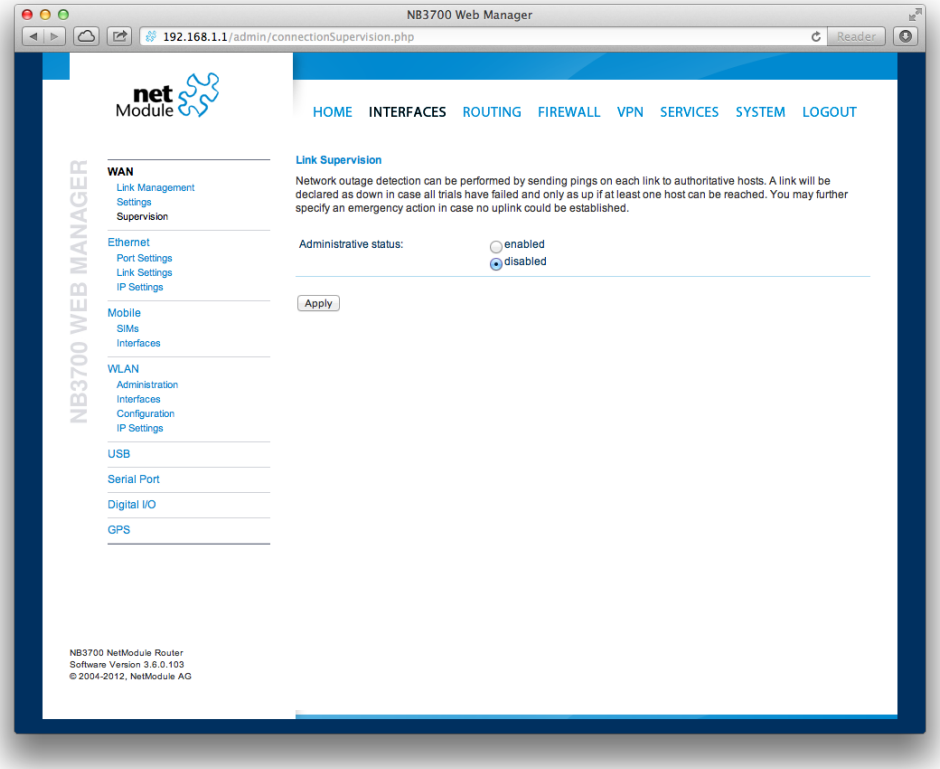


Figure 5.4.: Link Supervision

Administrative status: Enable or disable link supervision

Primary host: The primary host which will be monitored

Secondary host: The secondary host which will be monitored (optional)

Ping timeout: The amount of time in milliseconds a response for a single ping can take. Consider to increase this value in case of slow and tardy links (such as 2G connections)

Ping interval: The interval in seconds at which pings are transmitted on each interface

Max. number of failed trials: The maximum number of failed ping trials until the ping check will be declared as failed

You may further specify an emergency action in case no uplink can be established at all. You can choose between rebooting the system or restarting all involved link services

after a specific maximum downtime is reached.

Emergency action: The emergency action which should be taken after max. downtime reached. Using `reboot` would perform a reboot of the system, `restart services` will restart all link-related applications including a reset of the modems.

Maximum downtime: The maximum amount of downtime in minutes for which no WAN link could be established

5.3.2. Ethernet

Port Settings

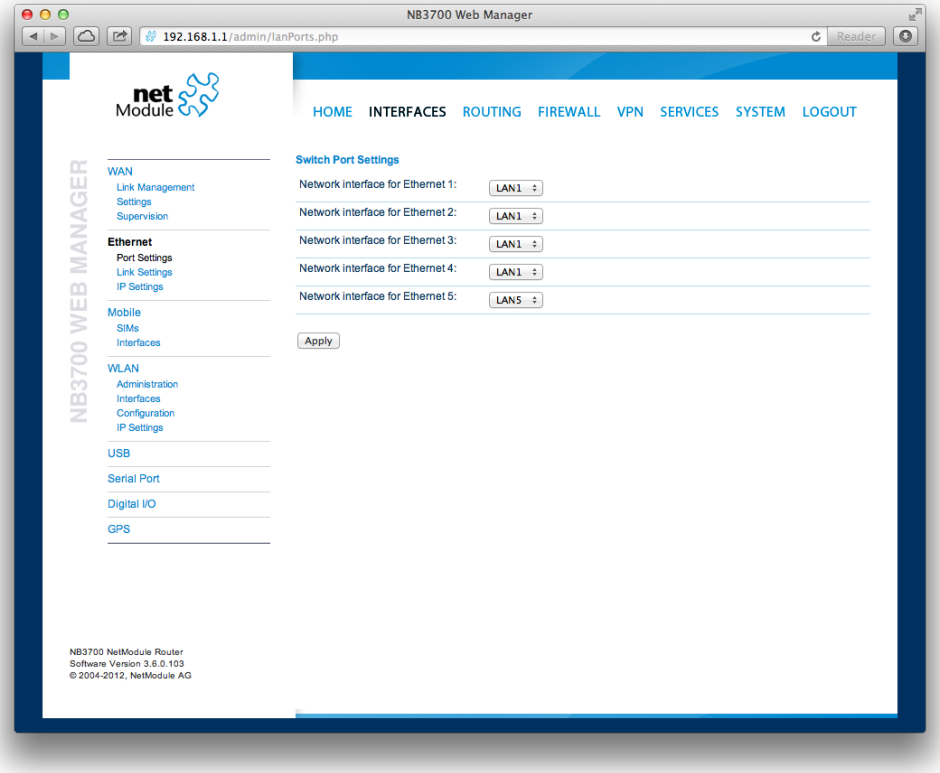


Figure 5.5.: LAN Ports

This menu can be used to individually assign each Ethernet port to a LAN interface, just in case you want to have different subnets per port or use one port as WAN interface. You may assign multiple ports to the same interface. Please note that on systems without an Ethernet switch, the ports will be bridged by software then and operated by running the Spanning Tree Protocol (STP).

Link Settings

Link negotiation can be set for each Ethernet port individually. Most devices support auto-negotiation which will configure the link speed automatically to comply with other devices in the network. In case of negotiation problems, you may assign the modes manually but it has to be ensured that all devices in the network utilize the same settings then.

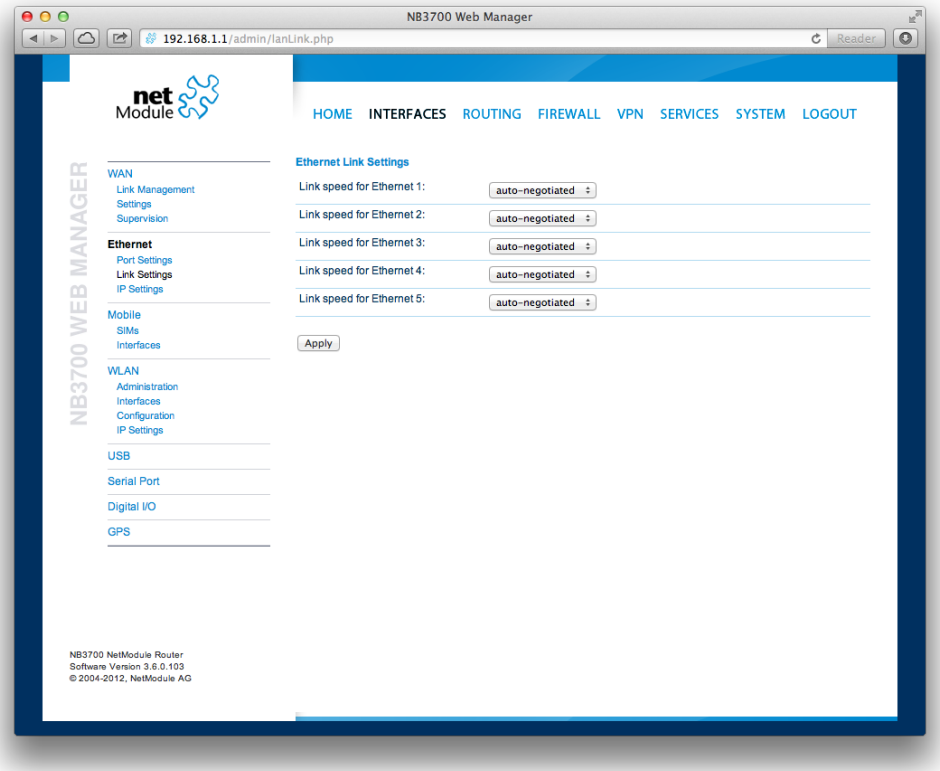


Figure 5.6.: LAN Link Settings

IP Settings

This page can be used to configure IP addressing for your LAN/WAN Ethernet interfaces.

Please keep in mind that the DNS servers can be set globally in the DNS server configuration menu. But as soon as a link comes up it will use the interface-specific nameservers (e.g. the ones being retrieved over DHCP) and update the resolver configuration accordingly.

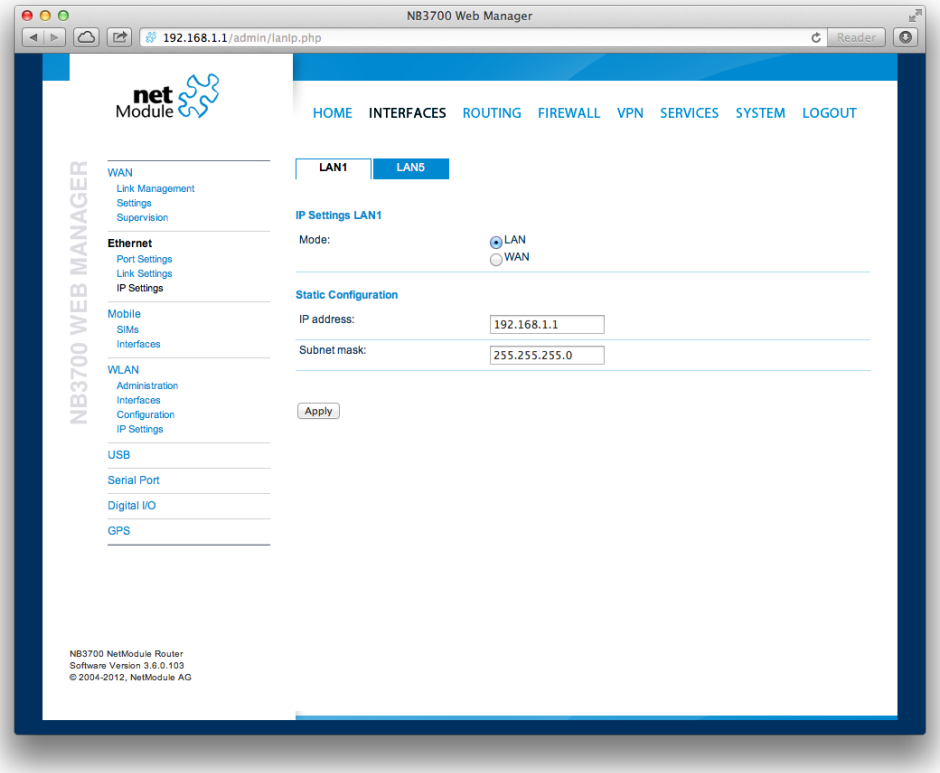


Figure 5.7.: LAN IP Configuration

Mode: Defines whether this interface is being used as LAN or WAN interface

When running in LAN mode the interface accepts the following static settings:

IP address: The IP interface address

Subnet mask: The subnet mask for this interface

When running in WAN mode the interface may be configured with the following settings:

WAN mode: The WAN operation mode, defines whether the interface should run as DHCP client, statically configured or over PPPoE

MTU: The maximum transfer unit for the interface, if provided it will specify the largest size of a packet

When running as DHCP client, no further configuration is required because all IP-related settings (address, subnet, gateway, DNS server) will be retrieved from a DHCP server in the network. You may also define static values but caution has to be taken to assign an unique IP address as it would otherwise raise IP conflicts in the network.

PPPoE is commonly used when communicating with another WAN access device (like a DSL modem). The following settings can be applied:

User name: PPPoE user name for authenticating at the access device

Password: PPPoE password for authenticating at the access device

Service name: Specifies the service name set of the access concentrator and can be left blank unless you have multiple services on the same physical network and need to specify the one you want to connect to

Access concentrator name: The name of the concentrator (the PPPoE client will connect to any access concentrator if left blank)

5.3.3. Mobile

SIMs

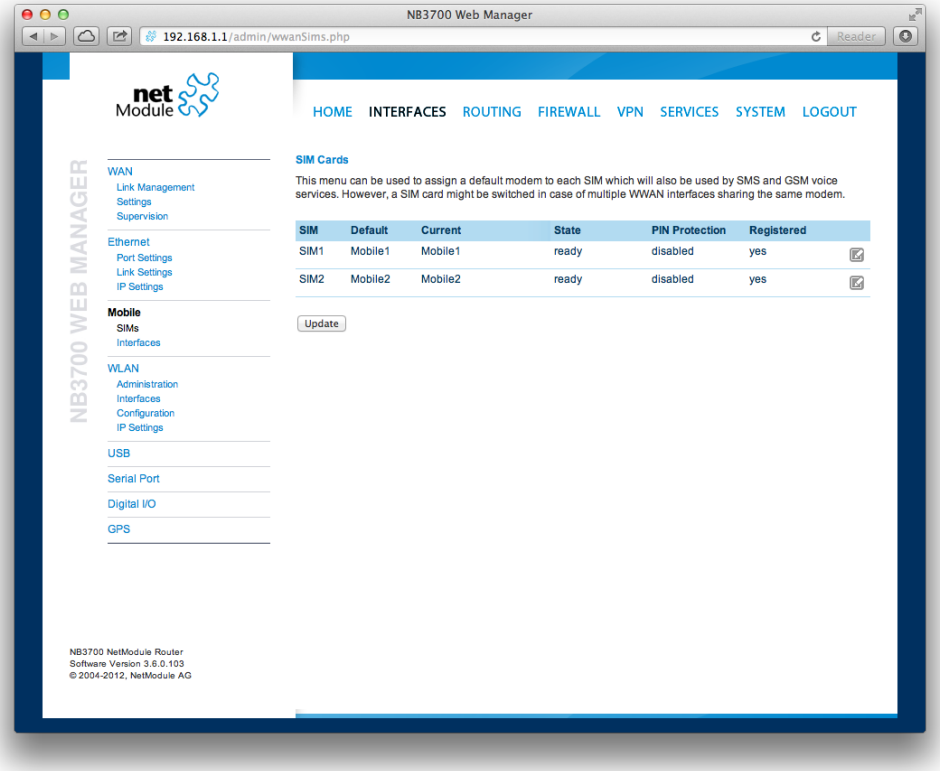


Figure 5.8.: SIMs

The SIM page gives an overview about the available SIM cards, their assigned modems and the current state. Once a SIM card has been inserted, assigned to a modem and successfully unlocked, the card should remain in state `ready` and the network registration status should have turned to `registered`. If not, please double-check your PIN.

Please keep in mind that registering to a network usually takes some time and depends on signal strength and possible radio interferences. You may hit the `Update` button at any time in order to restart PIN unlocking and trigger another network registration attempt.

Under some circumstances (e.g. in case the modem flaps between base stations) it might be necessary to set a specific service type or assign a fixed operator. The list of operators around can be obtained by initiating a network scan (may take up to 60 seconds). Further details can be retrieved by querying the modem directly, a set of suitable commands can be provided on request.

Configuration

A SIM card is generally assigned to a default modem but might be switched, for instance if you set up two WWAN interfaces with one modem but different SIM cards.

Close attention has to be paid when other services (such as SMS or Voice) are operating on that modem, as a SIM switch will naturally affect their operation.

The following settings can be applied:

Default modem: The default modem assigned to this SIM card

Service type: The service type to be used by default with this SIM card. Remember that the link manager might change this in case of different settings. The default is to use `automatic`, in areas with interfering base stations you can force a specific type (e.g. 3G-only) in order to prevent any flapping between the stations around.

PIN protection: Depending on the used card, it can be necessary to unlock the SIM with a PIN code. Please check the account details associated with your purchased SIM and figure out whether it is protected with a PIN.

PIN code: The PIN code for unlocking the SIM card

SMS gateway: The service center number for sending short messages. It is generally retrieved automatically from your SIM card but you may define a fix number here.

Network

This page provides information about the current network status, signal strength and the Local Area Identifier (LAI) to which the modem has been registered. An LAI is a globally unique number that identifies the country, network provider and Local Area Code (LAC, group of base stations) of any given location area. It can be used to force the modem to register to a particular mobile cell in case of competing stations.

You may further initiate a mobile network scan for getting networks in range and assign an LAI manually.

Query

This page allows you to send Hayes AT commands to the modem. Besides the 3GPP-conforming AT command-set further modem-specific commands can be applicable which we can provide on demand. Some modems also support running Unstructured Supplementary Service Data (USSD) requests, e.g. for querying the available balance of a prepaid account.

WWAN Interfaces

This page can be used to manage your WWAN interfaces. The resulting link will pop up automatically as WAN link once an interface has been added. Please refer to chapter 5.3.1 for how to manage them.

The Mobile LED will be blinking during the connection establishment process and goes on as soon as the connection is up. Refer to section 5.8.5 or consult the system log files for troubleshooting the problem in case the connection did not come up.

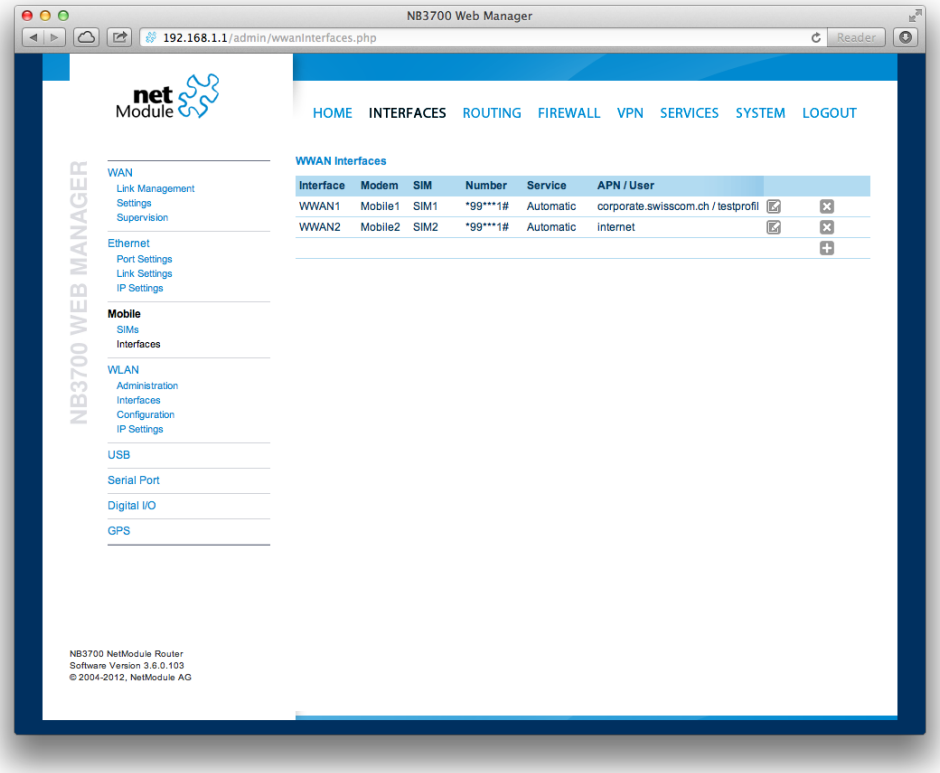


Figure 5.9.: WWAN Interfaces

The following mobile settings are required:

Modem: The modem to be used for this WWAN interface

SIM: The SIM card to be used for this WWAN interface

Service type: The required service type

Please note that these settings supersede the general SIM based settings as soon as the link is being dialed.

Generally, the connection settings are derived automatically as soon as the modem has registered and the network provider has been found in our database. Otherwise, it will be required to configure the following settings manually:

Phone number: The phone number to be dialed, for 3G+ connections this commonly refers to be *99***1#. For circuit-switched 2G connections you can enter the fixed phone number to be dialed in international format (e.g. +41xx).

Access point name: The access point name (APN) being used

Authentication: The authentication scheme being used, if required this can be PAP or/and CHAP

Username: The user-name used for authentication

Password: The password used for authentication

Furtheron, you may configure the following advanced settings:

Required signal strength: Sets a minimum required signal strength before the connection is dialed

Home network only: Determines whether the connection should only be dialed when registered to a home network

Negotiate DNS: Specifies whether the DNS negotiation should be performed and the retrieved name-servers should be applied to the system

Call to ISDN: Has to be enabled in case of 2G connections talking to an ISDN modem

IP header compression: Enables or disables Van Jacobson TCP/IP Header Compression for PPP-based connections. This feature will improve TCP/IP performance over slow serial links. Has to be supported by your provider.

Software compression: Enables or disables data compression for PPP-based connections. Software compression reduces the size of packets to improve throughput. Has to be supported by your provider.

Client address: Specifies a fixed client IP address on the mobile interface

MTU: The Maximum Transmission Unit represents the largest amount of data that can be transmitted within one IP packet and can be defined for any WAN interface.

5.3.4. WLAN

WLAN Management

In case your router is shipping with a WLAN (or Wi-Fi) module you can operate it either as `client` or `access point`. As a `client` it can create an additional WAN link which for instance can be used as backup link. As access point, it can form another LAN interface which can be either bridged to an Ethernet-based LAN interface or create a self-contained IP interface which can be used for routing and to provide services (such as DHCP/DNS/NTP) in the same way like an Ethernet LAN interface does.

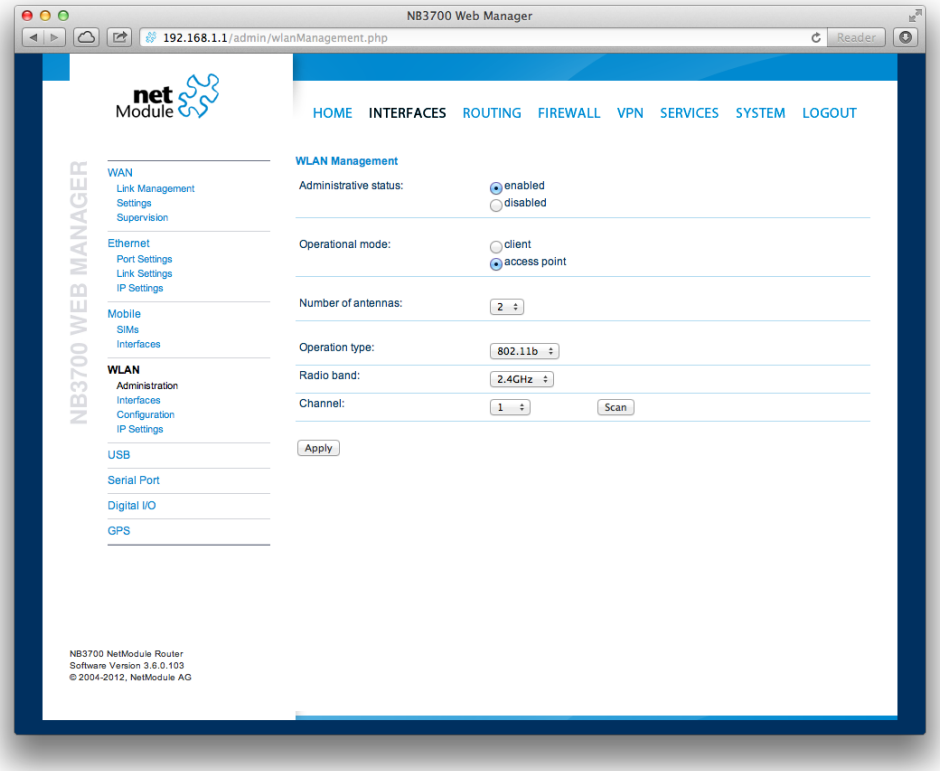


Figure 5.10.: WLAN Management

If the administrative status is set to `disabled`, the module will be powered off in order to reduce the overall power consumption. Regarding antennas, we generally recommend using two antennas for better coverage and throughput. A second antenna is definitely mandatory if you want to achieve higher throughput rates in 802.11n.

A WLAN `client` will automatically became a WAN link and can be managed as described in chapter 5.3.1.

Running as access point, you can further configure the following settings:

Operation type: Specifies the desired IEE 802.11 operation mode, 802.11a can be used

in the 5 GHz band, higher throughput in 20/40 MHz mode can be achieved with 802.11n

Radio band: Selects the radio band to be used for connections, depending on your module it could be 2.4 or 5 GHz

Channel: Specifies the channel to be used

Prior to setting up an `access point`, it is always a good idea to run a network scan for getting a list of neighboring WLAN networks and then choose the less interfering channel. Please keep in mind that two adequate channels are required for getting good throughputs with 802.11n in the 40 MHz radio band.

Running in `client` mode, you can select the network to which you want to connect to and enter the required authentication settings. You may also perform a WLAN network scan and pick the settings from the discovered information directly. The credentials can be obtained by the operator of your WLAN `access point`.

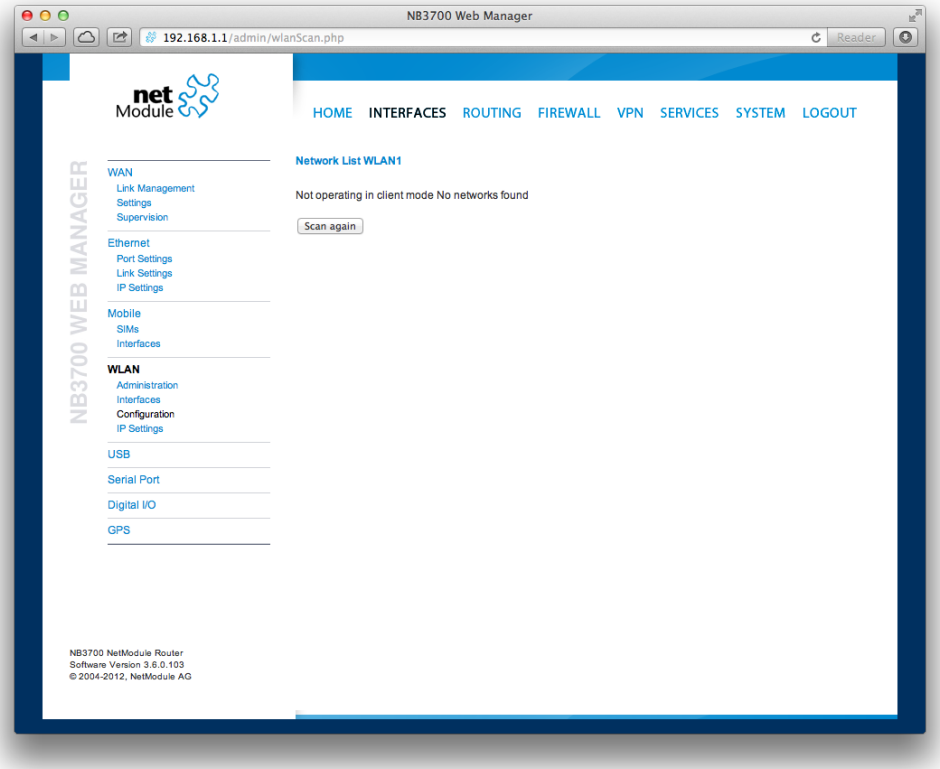


Figure 5.11.: WLAN Scan

WLAN Interfaces

An `access point` can define up to 4 networks being broadcasted. The networks can be individually bridged to a LAN interface or operate as dedicated interface in routing-mode.

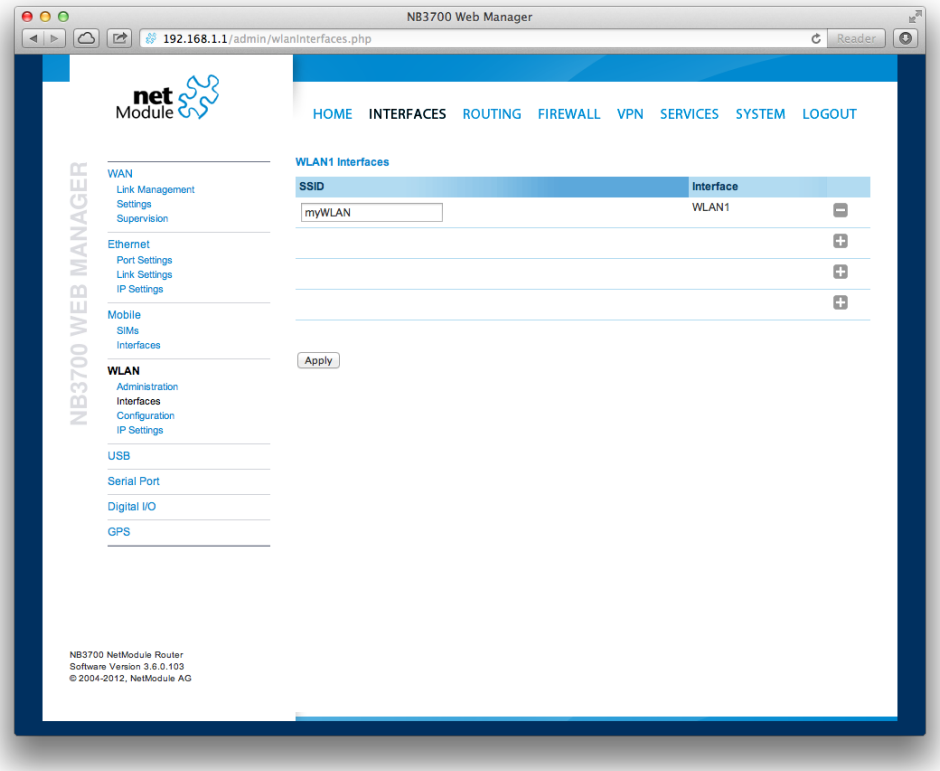


Figure 5.12.: WLAN Interfaces

WLAN Configuration

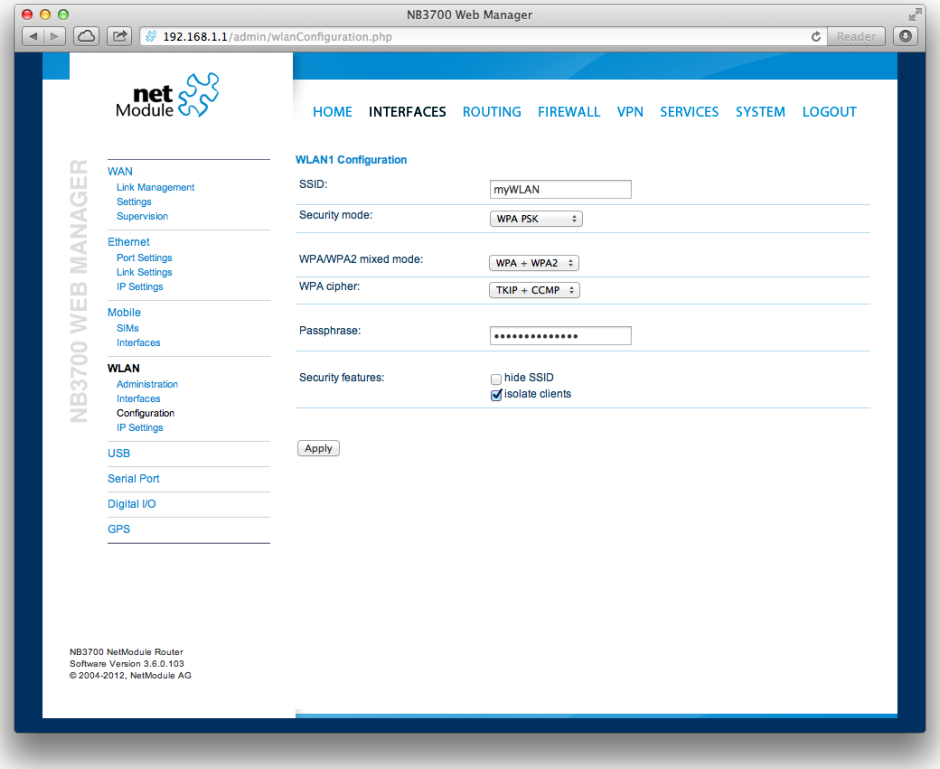


Figure 5.13.: WLAN Configuration

Running in `access point` mode you can define up to 4 SSIDs with each running their own network configuration. This section can be used to configure security-related settings.

SSID: The network name (called SSID)

Security mode: The desired security mode (such as WPA PSK), WPA (802.1x) can be used to authenticate against a remote RADIUS server which can be configured in chapter 5.8.2

WPA/WPA2 mixed mode: WPA2 should be preferred over WPA1, running WPA/WPA2 mixed-mode offers both

WPA cipher: The WPA cipher to be used, the default is to run both (TKIP and CCMP)

Passphrase: The passphrase used for authentication

WLAN IP Settings

This section lets you configure the TCP/IP settings of your WLAN network.

A `client` interface can be run over DHCP or with a statically configured address and default gateway.

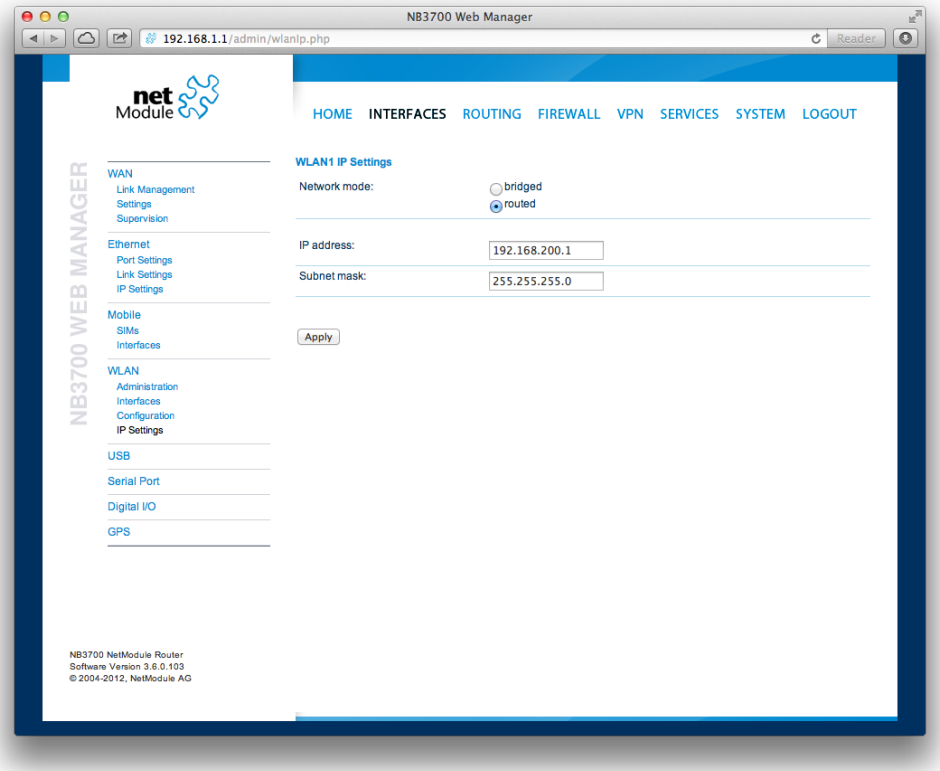


Figure 5.14.: WLAN IP Configuration

The `access point` networks can be bridged to any LAN interface for letting WLAN clients and Ethernet hosts operate in the same subnet. However, for multiple SSIDs we strongly recommend to set up separated interfaces in routing-mode in order to avoid unwanted access and traffic between the interfaces. The corresponding DHCP server for each network can be configured in afterwards as described in chapter 5.7.2.

Network mode: Choose whether the interface shall be operated bridged or in routing-mode

Bridge interface: If bridged, the LAN interface to which the WLAN network should be bridged

IP address / netmask: In routing-mode, the IP address and netmask for this WLAN network

5.3.5. USB

NetModule routers ship with a standard USB 2.0 host port which can be used to connect any arbitrary USB device.

USB Autorun

This feature can be used to automatically launch a shell script or perform a software/-config update as soon as an USB storage stick has been plugged in. For authentication, a file called `autorun.key` must exist in the root directory of a FAT16/32 formatted stick. It can be downloaded from that page and corresponds to the SHA256 Hash of the admin password. The file can hold multiple hashes which will be processed line-by-line during authentication which can be used for setting up more systems with different admin passwords.

Once authentication has succeeded, the system scans for other files in the root directory which can perform the following actions:

1. For running a script: `autorun.sh`
2. For a configuration update: `cfg-<SERIALNO>.zip` (e.g. `cfg-00112B000815.zip`)
3. For a software update: `sw-update.img`

USB Device Server

As soon as the USB device server has been enabled you can refresh the discovered USB devices plugged in and attach them to the USB/IP server. Enabled devices can now be exported to a remote host. You will need an additional driver on the remote site (a pre-packed Windows driver package can be obtained from http://www.netmodule.com/products) and further installation instructions which we will happily provide on demand.

Please note, that some USB devices behave latency-sensitive which may raise problems when run over a slow IP connection. In addition, it may also happen that the device driver is not able to work with the USB/IP driver. Please contact our support in case of compatibility issues.

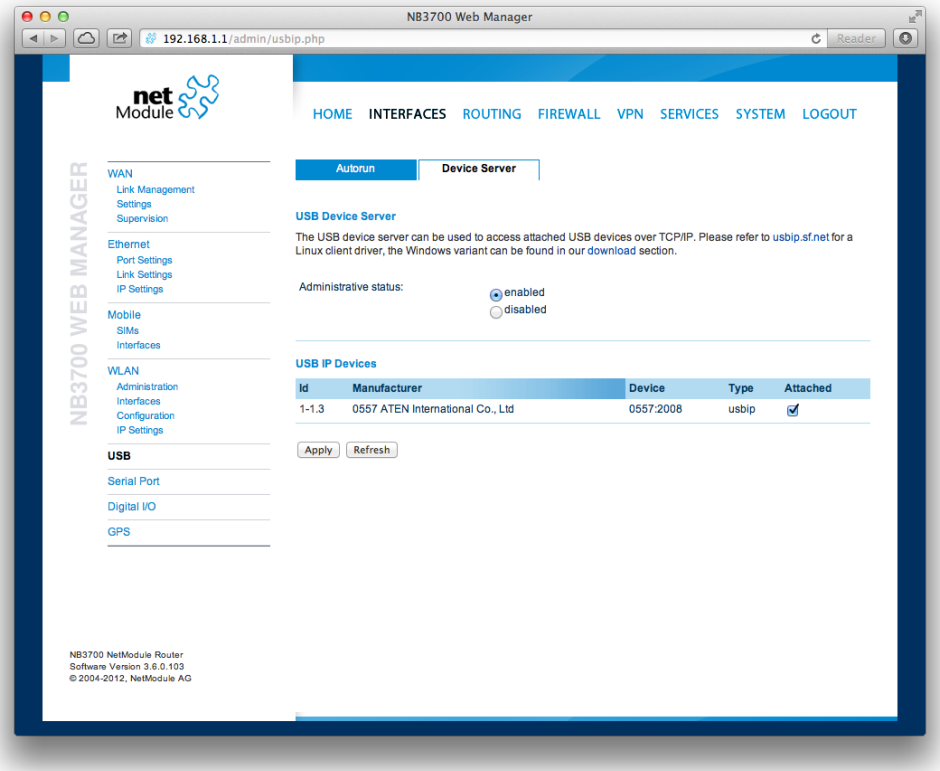


Figure 5.15.: USB Device Server

5.3.6. Serial Port

This page can be used to manage your serial ports. They can be used for various purposes on the system. When set to `none` it will be disabled, when set to `login console` you would be able to get a login shell when connecting to the serial port (`115200 8N1`). You may also mark them as reserved for SDK scripts.

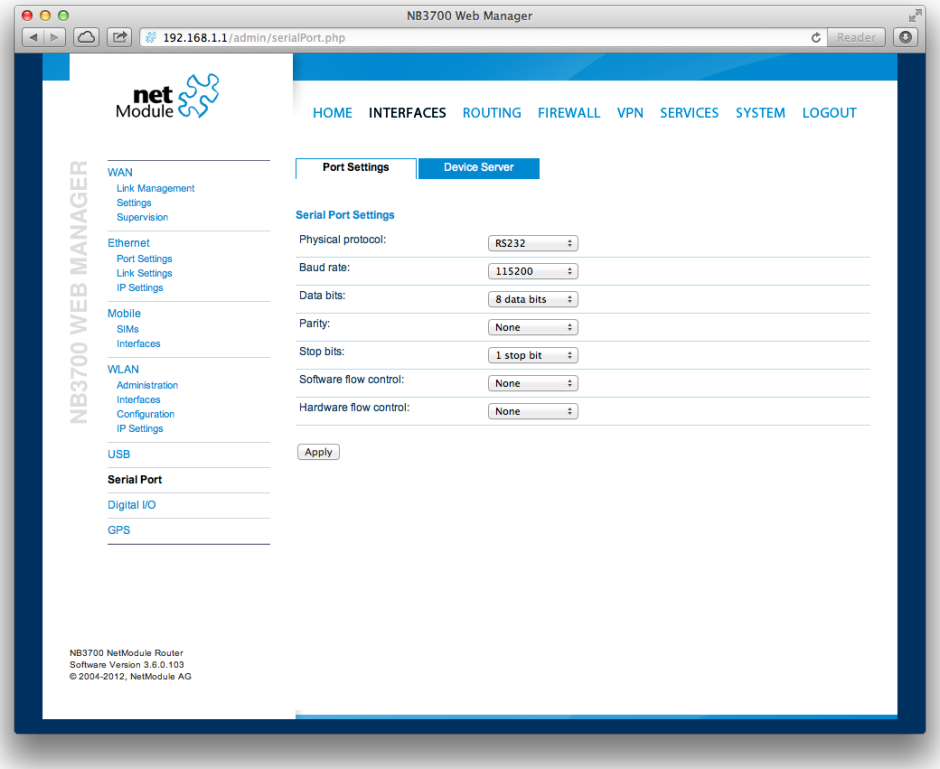


Figure 5.16.: Serial Port

Furtheron, a device server can be run for each port which can be used to control the serial device via IP.

It can be configured as follows:

Physical protocol: Selects the desired physical protocol on the serial port

Baud rate: Specifies the baud rate run on the serial port

Data bits: Specifies the number of data bits contained in each frame

Parity: Specifies the parity used for every frame that is transmitted or received

Stop bits: Specifies the number of stop bits used to indicate the end of a frame

Software flow control: Defines the software flow control for the serial port, `XOFF` will send a stop, `XON` a start character to the other end to control the rate of any incoming data

Hardware flow control: You may enable RTS/CTS hardware flow control, so that the RTS and CTS lines are used to control the flow of data

Protocol on TCP/IP: You may choose the IP protocols `Telnet` or `TCP raw` for the device server

Port: The TCP port for the device server

Timeout: The timeout until a client is declared as disconnected

The Serial Port is optional on NB3700.

5.3.7. Digital I/O

The Digital I/O page displays the current status of the I/O ports and can be used to turn output ports `on` or `off`.
You can apply the following settings:

DO1 after reboot: Initial status of DO1 after system has booted
DO2 after reboot: Initial status of DO2 after system has booted

Besides `on` and `off` you may keep the `default` status as the hardware has initialized it after power-up.
The digital inputs and outputs can also be monitored and controlled by SDK scripts.

5.3.8. GPS

Administration

The GPS page lets you enable or disable the GPS modules present in the system and can be used to configure the daemon that can be used to share access to receivers without contention or loss of data and to respond to queries with a format that is substantially easier to parse than the NMEA 0183 emitted directly by the GPS device.

We are currently running the Berlios GPS daemon (version 2.37), please navigate to `http://gpsd.berlios.de` for getting more information about how to incorporate it. The GPS values can also be queried by the CLI and used in SDK scripts.

Administrative status: Enable or disable GPS reception

Antenna type: The type of the connected GPS antenna, either active or passive

Server port: The TCP port on which the daemon is listening for incoming connections

Allow clients from: Specifies where clients can connect from, can be either `everywhere` or from a specific network

Clients start mode: Specifies how client reception is started upon connect. You can specify `on request`, which typically requires an `R` to be sent, or `raw/super-raw` mode which will transmit NMEA frames to the client instantly

Position

This page shows the current position of the box together with a location map.

Information

This pages provides further information about the satellites in view and values derived from them:

Latitude: The geographic coordinate specifying the north-south position

Longitude: The geographic coordinate specifying the east-west position

Altitude: The height above sea level of the current location

Satellites in view: The number of satellites in view as stated in GPGSV frames

Speed: The horizontal and vertical speed in meter per second as stated in GPRMC frames

Satellites used: The number of satellites used for calculating the position as stated in GPGGA frames

Dilution of precision: The dilution of precision as stated in GPGSA frames

Furtheron, each satellite also comes with the following details:

PRN: The PRN code of the satelitte (also referred as satellite ID) as stated in GPGSA frames

Elevation: The elevation (up-down angle between the dish pointing direction) in degrees as stated in GPGSV frames

Azimuth: The azimuth (rotation around the vertical axis) in degrees as stated in GPGSV frames

SNR: The SNR (Signal to Noise Ratio), often referred as signal strength

Please note that the values are shown as calculated by the daemon, their accuracy might be suggestive.

5.4. ROUTING

5.4.1. Static Routes

This menu shows all routing entries of the system. They are typically formed by an address/netmask couple (represented in IPv4 dotted decimal notation) which specify the destination of a packet. The packets can be directed to either a gateway or an interface or both. If interface is set to ANY, the system will choose the route interface automatically, depending on the best matching network configured for an interface.

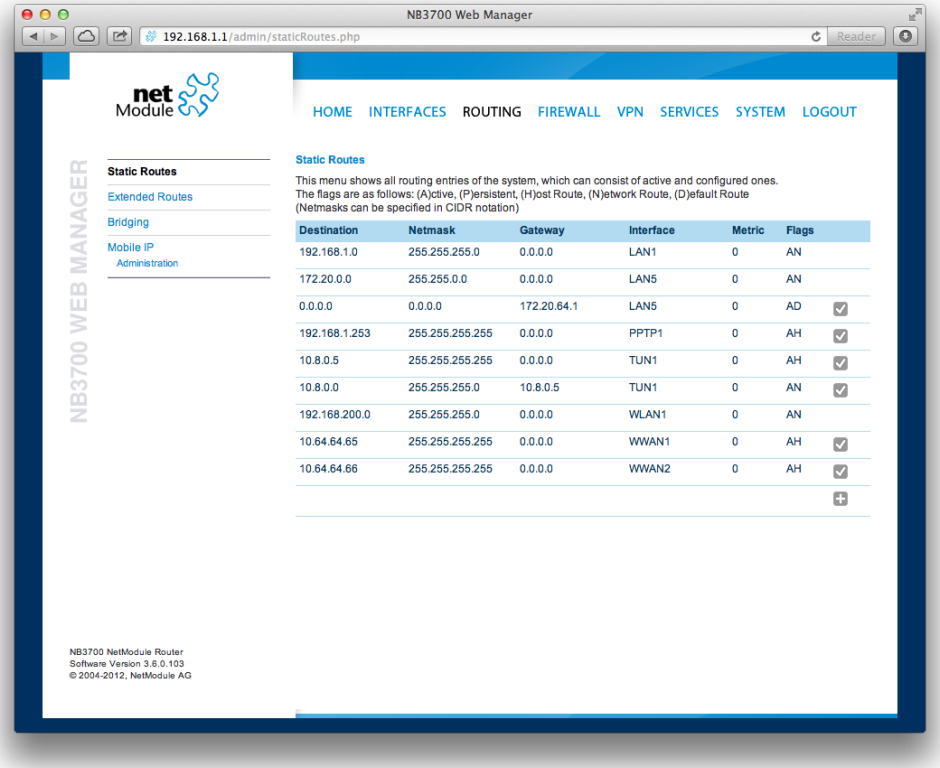

NB3700 Web Manager
192.168.1.1/admin/staticRoutes.php
Reader
net Module
HOME INTERFACES ROUTING FIREWALL VPN SERVICES SYSTEM LOGOUT
NB3700 WEB MANAGER
Static Routes
Extended Routes
Bridging
Mobile IP
Administration
Static Routes
This menu shows all routing entries of the system, which can consist of active and configured ones.
The flags are as follows: (A)ctive, (P)ersistent, (H)ost Route, (N)etwork Route, (D)efault Route
(Netmasks can be specified in CIDR notation)
Destination Netmask Gateway Interface Metric Flags
192.168.1.0 255.255.255.0 0.0.0.0 LAN1 0 AN
172.20.0.0 255.255.0.0 0.0.0.0 LAN5 0 AN
0.0.0.0 0.0.0.0 172.20.64.1 LAN5 0 AD
192.168.1.253 255.255.255.255 0.0.0.0 PPTP1 0 AH
10.8.0.5 255.255.255.255 0.0.0.0 TUN1 0 AH
10.8.0.0 255.255.255.0 10.8.0.5 TUN1 0 AN
192.168.200.0 255.255.255.0 0.0.0.0 WLAN1 0 AN
10.64.64.65 255.255.255.255 0.0.0.0 WWAN1 0 AH
10.64.64.66 255.255.255.255 0.0.0.0 WWAN2 0 AH
NB3700 NetModule Router
Software Version 3.6.0.103
© 2004-2012, NetModule AG

Figure 5.17.: Static Routing

In general, host routes precede network routes and network routes precede default routes. Additionally, a metric can be used to determine the priority of a route, a packet will go in the direction with the lowest metric in case a destination matches multiple routes.

Netmasks can be specified in CIDR notation (i.e. /24 expands to 255.255.255.0).

Destination: The destination address of a packet

Netmask: The subnet mask which forms, in combination with the destination, the network to be addressed. A single host can be specified by a netmask of 255.255.255.255, a default route corresponds to 0.0.0.0.

Gateway: The next hop which operates as gateway for this network (can be omitted on peer-to-peer links)

Interface: The network interface on which a packet will be transmitted in order to reach the gateway or network behind it

Metric: The routing metric of the interface (default 0), higher metrics have the effect of making a route less favorable

Flags: (A)ctive, (P)ersistent, (H)ost Route, (N)etwork Route, (D)efault Route

The flags obtain the following meanings:

Flag	Description
A	The route is considered active, it might be inactive if the interface for this route is not yet up.
P	The route is persistent, which means it is a configured route, otherwise it corresponds to an interface route.
H	The route is a host route, typically the netmask is set to 255.255.255.255.
N	The route is a network route, consisting of an address and netmask which forms the subnet to be addressed.
D	The route is a default route, address and netmask are set to 0.0.0.0, thus matching any packet.

Table 5.2.: Static Route Flags

5.4.2. Extended Routing

Extended routes can be used to perform policy-based routing, they generally precede static routes.

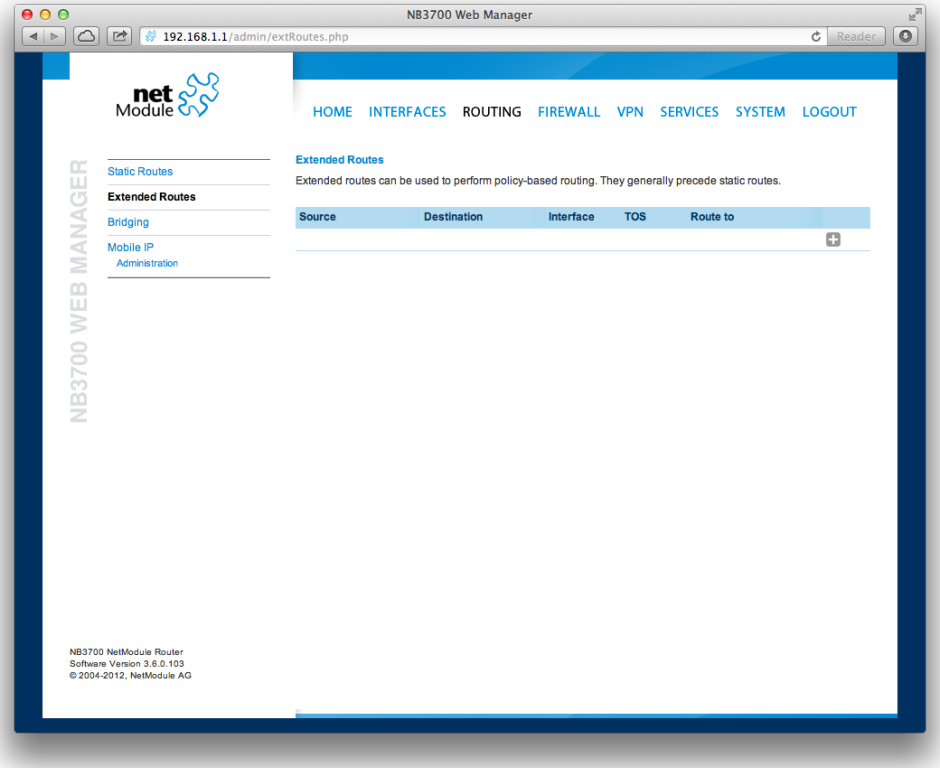


Figure 5.18.: Extended Routing

In contrast to statis routes, extended routes can be made up, not only of a destination address/netmask, but also a source address/netmask, incoming interface and the type of service (TOS) of packets.

Source address: The source address of a packet

Source netmask: The source address of a packet

Destination address: The destination address of a packet

Destination netmask: The destination address of a packet

Incoming interface: The interface on which the packet enters the system

Type of service: The TOS value within the header of the packet

Route to: Specifies the target interface or gateway to where the packet should get routed to

5.4.3. Bridging

This menu shows the currently configured bridge groups and their interface members. Please refer to section A.1 for a list of interface abbrevations.

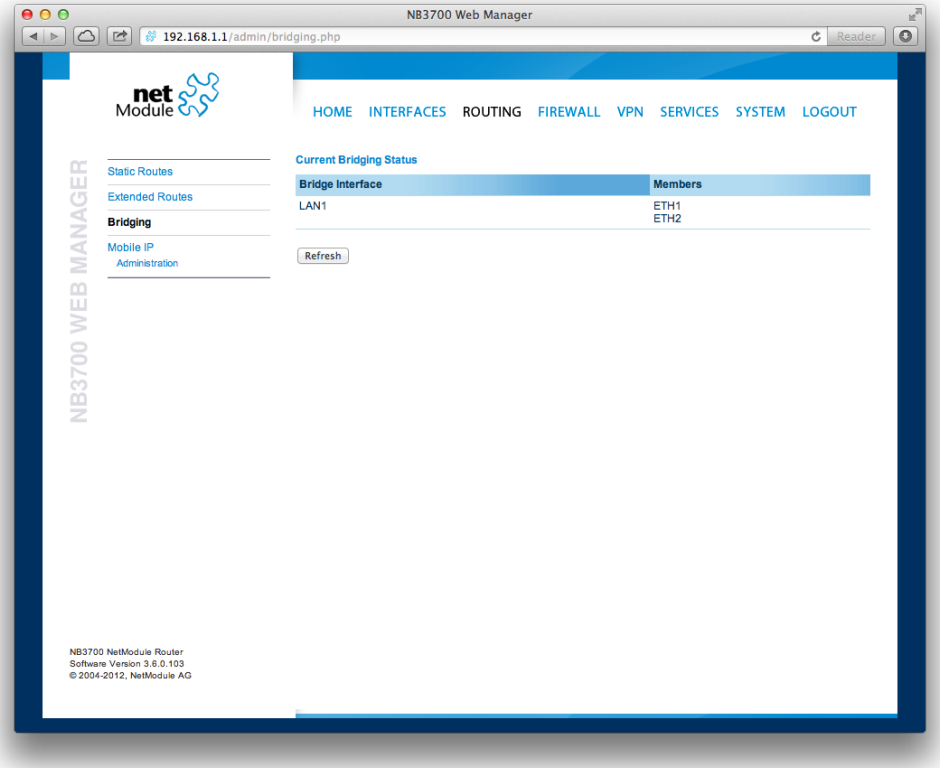


Figure 5.19.: Bridging

5.4.4. Mobile IP

Mobile IP (MIP) can be used to enable seamless switching between different kinds of WAN links (e.g. WWAN/WLAN). The `mobile node` hereby remains reachable via the same IP address (`home address`) at any time, independently of the WAN link being used. Effectively, any WAN link switch causes very small outages during switchover while keeping all IP connections alive.

Moreover, NetModule routers also support NAT-Traversal for mobile nodes running behind a firewall (performing NAT), which makes mobile nodes even there accessible from a central office via their home address, and thus, bypassing any complicated VPN setups.

The `home agent` accomplishes this by establishing a tunnel (similar to a VPN tunnel) between itself and the `mobile node`. WAN link switching works by telling the `home agent` that the WAN IP address (called the `care-of address` in MIP terms) of the `mobile node` has changed. The `home agent` will then encapsulate packets destined to a `mobile node`'s home address into a tunnel packet containing the current `care-of address` of the `mobile node` as its destination address.

To prevent problems with firewalls and private IP addressing, the MIP implementation always employs reverse tunneling, which means that all traffic sent by a `mobile node` is relayed via the tunnel to the `home agent` instead of directly being conveyed to the final destination. This fact also empowers MIP to be used as a lightweight VPN replacement (without payload secrecy).

The MIP implementation supports RFCs 3344, 5177, 3024 and 3519. For applications requiring vast numbers of mobile nodes, interoperability with the Cisco 2900 Series `home agent` implementation has been verified. However, since NetModule routers implement a `mobile node` as well as a `home agent`, a MIP network with up to 10 mobile nodes can be implemented without requiring expensive third party routers.

If MIP is run as a `mobile node`, the following settings can be configured:

Primary home agent address: The address of the primary `home agent`

Secondary home agent address: The address of the secondary `home agent`. The mobile node will try to register with this home agent, if the primary `home agent` is not reachable.

Home address: The permanent home address of the `mobile node` which can be used to reach the mobile router at any time

SPI: The Security Parameter Index (SPI) identifying the security context for the mobile IP tunnel between the `mobile node` and the `home agent`. This is used to distinguish mobile nodes from each other. Therefore each mobile node needs to be assigned a unique SPI. This is a 32-bit hexadecimal value.

Authentication type: The used authentication algorithm. This can be prefix-suffix-md5 (default for MIP) or hmac-md5.

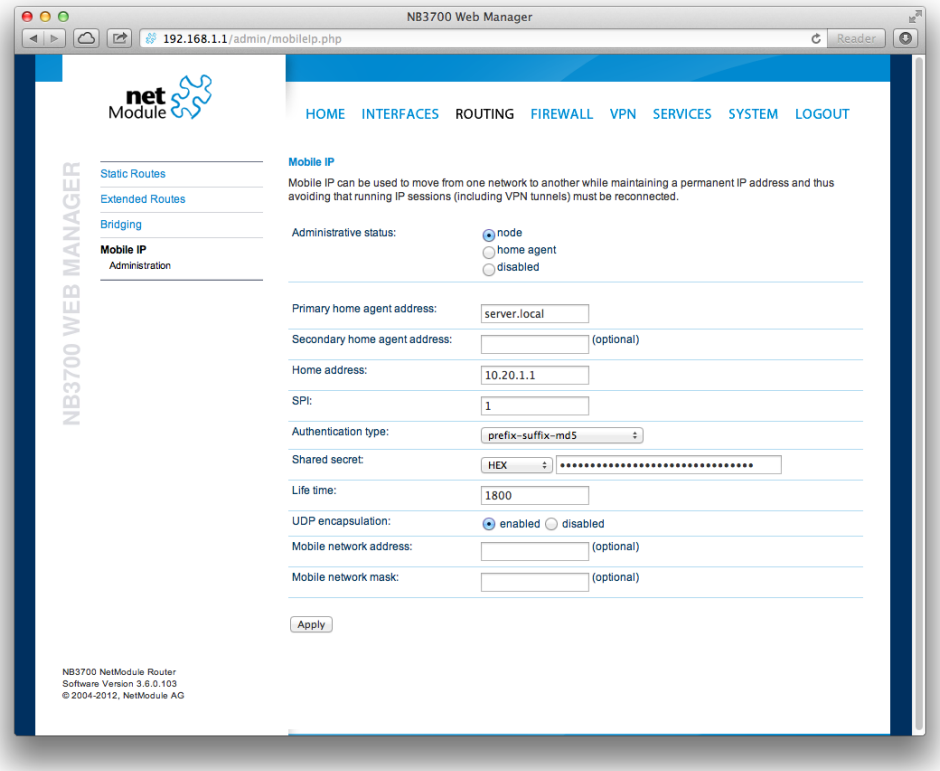


Figure 5.20.: Mobile IP

Shared secret: The shared secret used for authentication of the `mobile node` at the `home agent`. This can be either a 128-bit hexadecimal value or a random length ASCII string.

Life time: The lifetime of security associations in seconds

UDP encapsulation: Specifies whether UDP encapsulation shall be used or not. To allow NAT traversal, UDP encapsulation must be enabled.

Mobile network address: Optionally specifies a subnet which should be routed to the `mobile node`. This information is forwarded via Network Mobility (NEMO) extensions to the `home agent`. The `home agent` can then automatically add IP routes to the subnet via the `mobile node`. Note that this feature is not supported by all third party `home agent` implementations.

Mobile network mask: The network mask for the optional routed network

If MIP is run as a `home agent`, you will have to set up a home address and network mask for the `home agent` first. Then you will need to add the configuration for all mobile nodes, which is made up of the following settings:

SPI: The Security Parameter Index (SPI) identifying the security context for the tunnel between the `mobile node` and the `home agent`. This is used to distinguish mobile nodes from each other. Therefore each `mobile node` needs to be assigned a unique SPI. This is a 32-bit hexadecimal value.

Authentication type: The used authentication algorithm. This can be prefix-suffix-md5 (default for mobile IP) or hmac-md5.

Shared secret: The shared secret used for authentication of the `mobile node` at the `home agent`. This can be either a 128-bit hexadecimal value or a random length ASCII string.

5.5. FIREWALL

5.5.1. Administration

NetModule routers use Linux's netfilter/iptables firewall framework (see http://www.netfilter.org for more information) which supports stateful inspection, that is, granting the same permissions for inherited connections within an IP session (e.g. FTP which builds up a control and data connection).
The administration page can be used to enable and disable firewalling. When turning it on, a shortcut can be used to generate a predefined set of rules which allow administration (over HTTP, HTTPS, SSH or TELNET) by default but block any other packets coming from the WAN interface.

5.5.2. Rules

In general, the firewall is set up of a range of rules which control each packet's permission to pass the router. Please note that the rules are processed by order, that means traversing the list from top to bottom until a matching rule is found. Packets which are not matching any of the rules configured will be ALLOWED.

Description: A meaningful description about the purpose of this rule

Mode: Specifies whether the packets of this rule should be allowed or denied

Source: The source address of matching packets, can be any or specified by an address/network

Destination: The destination address of matching packets, can be any, local (addressed to the system itself) or specified by an address/network

Incoming interface: The interface on which matching packets are received

Protocol: The used IP protocol of matching packets, can be UDP, TCP or ICMP

Destination port(s): The destination port of matching packets, which can be specified by a single port or a range of ports (only UDP/TCP)

The statistics page can be used to figure out if rules have matched any packets and provides a convenient way to debug your firewall setup.

5.5.3. NAPT

This page can be used to configure Network Address and Port Translation (NAPT) for packets traversing the system. NAPT hereby modifies IP addresses or/and TCP/UDP ports in matching IP packets. By tracking those connections, it will also automatically adjust the returning packets of an IP session.

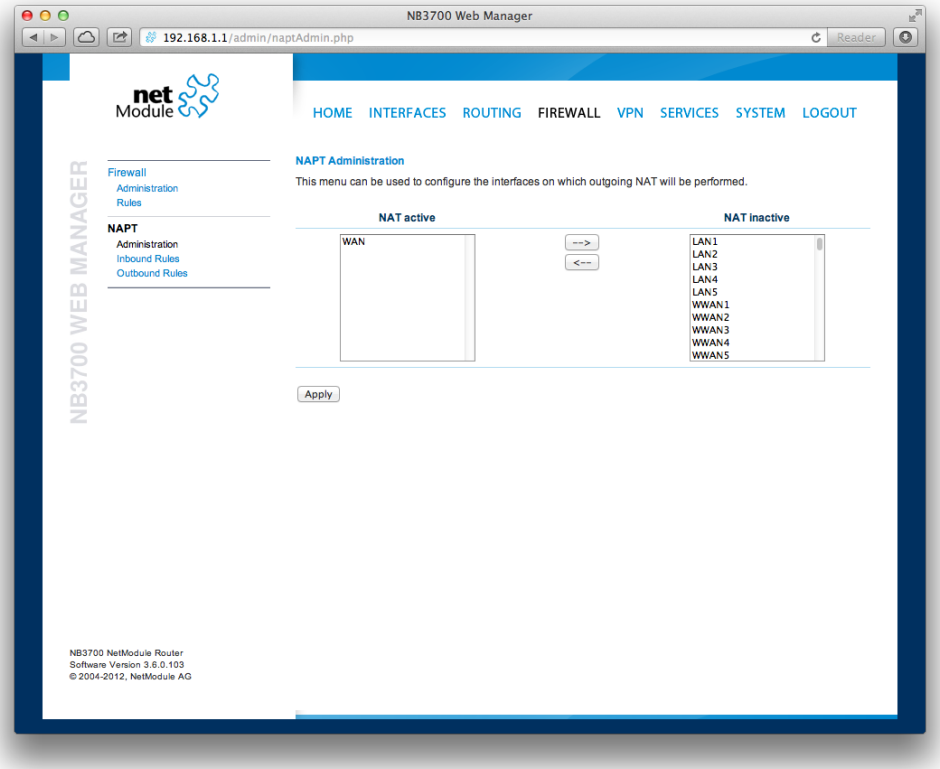


Figure 5.21.: NAPT Administration

The administration page lets you specify the interfaces on which outgoing NAT (also called *Masquerading*) will be performed. NAT will hereby use the address of the selected interface and choose a random source port for outgoing connections and thus enables communication between hosts from a private local area network towards hosts on the public network.

NAPT Inbound Rules

Inbound rules can be used to modify the target section of IP packets and, for instance, forward a service or port to an internal host. By doing so, you can expose that service and make it available from the Internet. You may also establish 1:1 NAT mapping for a single host using additional outbound rules.

Please note that the specified rules are processed by order, that means, traversing the

list from top to bottom until a matching rule is found. If there is no matching rule found, the packet will pass as is.

Description: A meaningful description of this rule

Incoming interface: The interface from which matching packets are received

Target address: The destination address of matching packets (optional)

Protocol: The used protocol of matching packets

Ports: The used UDP/TCP port of matching packets

Redirect to: The address to which matching packets shall be redirected

Redirect port: The port to which matching packets will be redirected

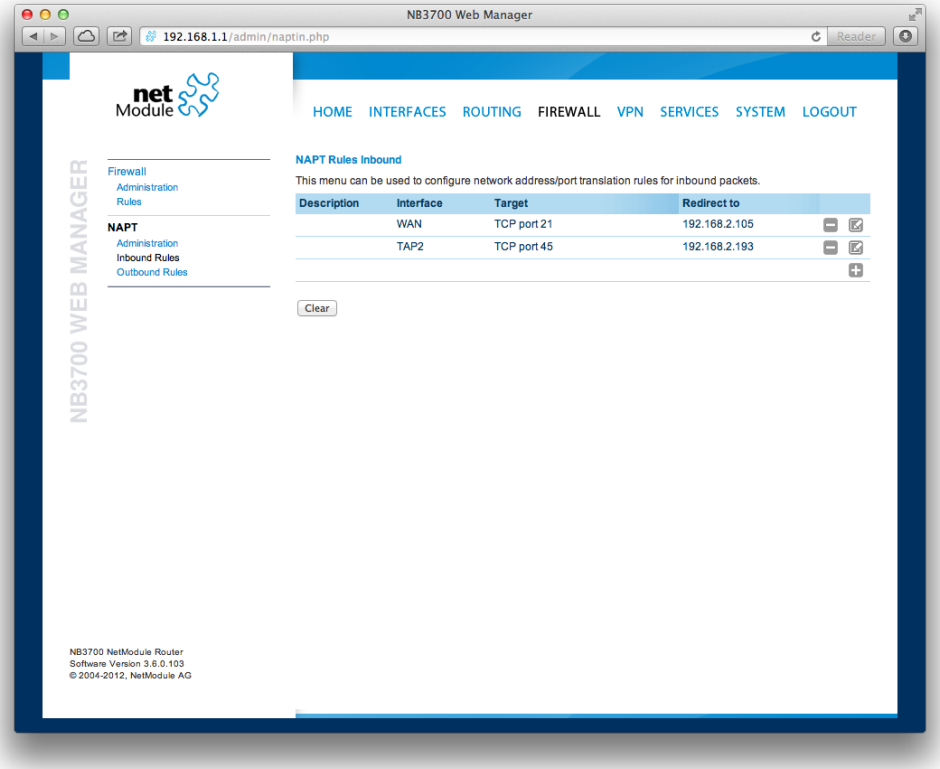


Figure 5.22.: Inbound NAPT

NAPT Outbound Rules

Outbound rules will modify the source section of IP packets and can be used to establish 1:1 NAT mappings but also to redirect packets to a specific service.

Description: A meaningful description of this rule

Incoming interface: The outgoing interface on which matching packets are leaving the router

Source address: The source address of matching packets (optional)

Protocol: The used protocol of matching packets

Ports: The used UDP/TCP port of matching packets

Rewrite source address: The address to which the source address of matching packets shall be rewritten

Rewrite source port: The port to which the source port of matching packets shall be rewritten

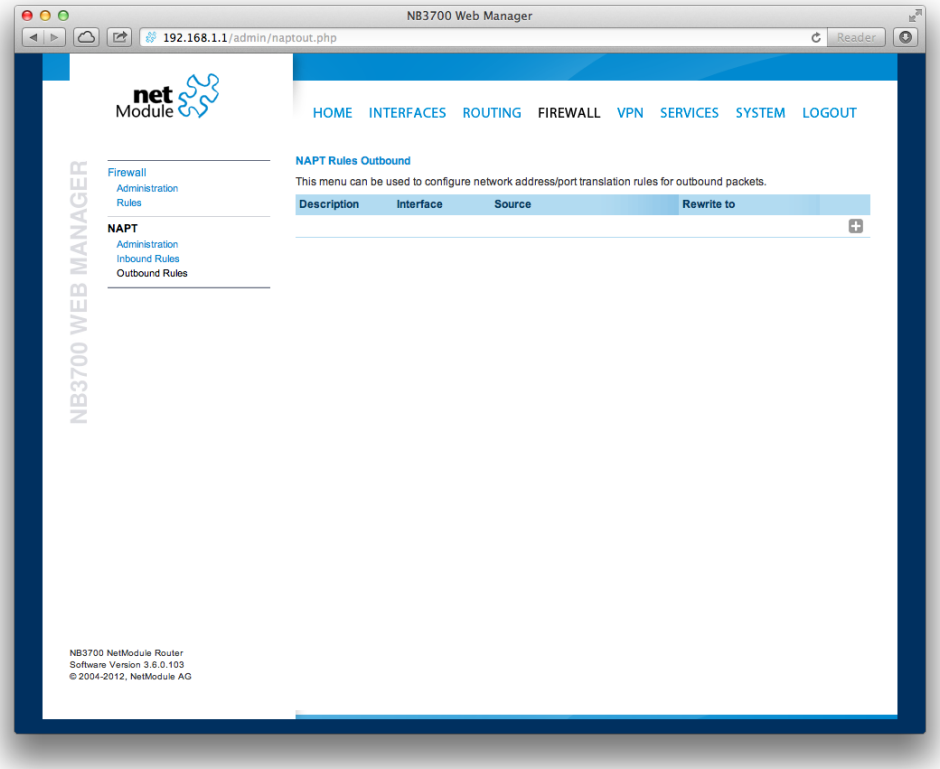


Figure 5.23.: Outbound NAPT

5.6. VPN

5.6.1. OpenVPN

OpenVPN Administration

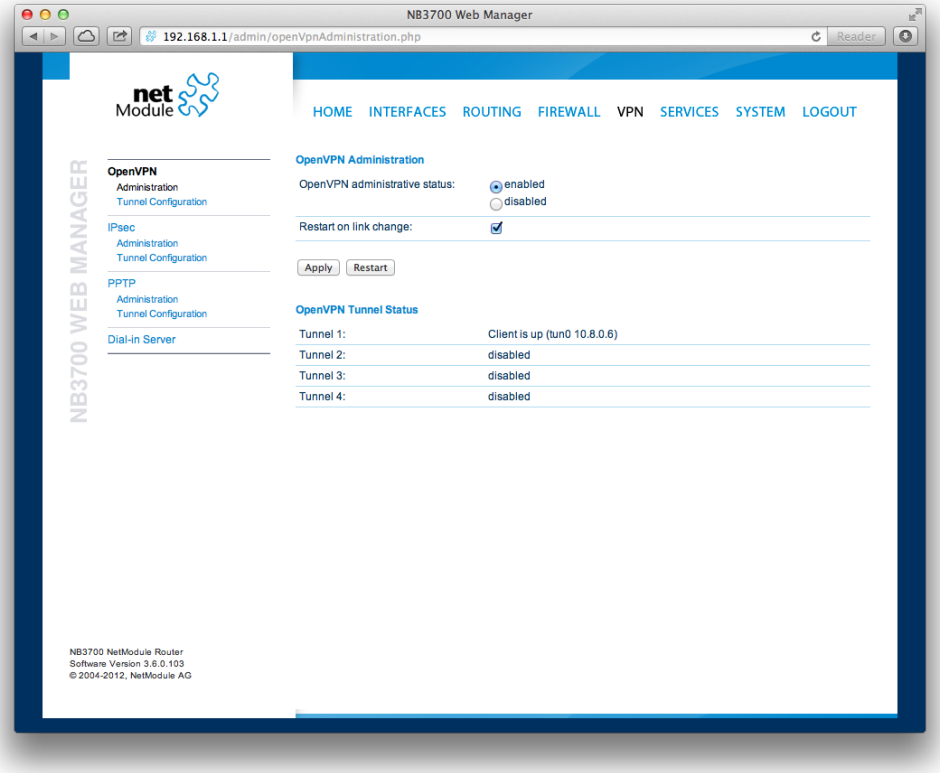


Figure 5.24.: OpenVPN Administration

Tunnel Configuration

NetModule routers support one single server tunnel and up to four client tunnels. You can specify tunnel parameters either in standard configuration or upload an expert mode file which has been created in advance. Refer to chapter 5.6.1 to learn more about how to manage clients and generate the files.

If the tunnel is operated in client mode, the following settings can be applied:

Operation mode: Specifies whether client or server mode should be used for this tunnel

Primary server address: The address of the primary server

Primary server port: The port of the primary server (1194 by default)

Secondary server address: The address of the secondary server address (optional) which will be used in case the primary address cannot be reached

Secondary server port: The port of the secondary server (optional)

Setting up a tunnel server just requires the server port to be set, the settings mentioned below apply for both, server and client tunnels:

Type: The encapsulation type for this tunnel which can be either TUN (typically used for routed connections) or TAP (needed for bridged networks)

Network mode: Defines how the packets should be forwarded, which can be either routed or bridged from/to a particular LAN interface

Cipher: The required cipher mechanism used for encryption

Use compression: Enable or disable packet compression

Use keepalive: Can be used to send a periodic keepalive packet in order to keep the tunnel up despite of inactivity

Redirect gateway: By redirecting the gateway, all packets will be directed to the VPN tunnel. Please ensure that essential services (such as DNS or NTP servers) can be reached at the network behind the tunnel. In doubt, create an extra static route pointing to the correct interface.

Protocol: The tunnel protocol to be used for the transport connection

Authentication: You can choose between no authentication, credential-based (where you have to specify a username and password) as well as based on keys and certificates. Note that keys/certificates have to be created or uploaded for making the tunnel work.

ExpertConfiguration

OpenVPN Expert Configuration (Client)

The expert configuration mode offers a straightforward way to configure a tunnel by simply uploading a package containing the required configuration and optionally key/certificate files. A client tunnel usually consists of the following files:

client.conf: OpenVPN configuration file (see `http://www.openvpn.net` for available options)

ca.crt: aoot certificate authority file

client.crt: Certificate file

client.key: Private key file

Please note that you may specify arbitrary file names, however, the configuration file suffix must be `.conf` and all files referred in the configuration file must correspond to relative path names.

OpenVPN Expert Configuration (Server) A server tunnel typically requires the following files:

server.conf: OpenVPN configuration file

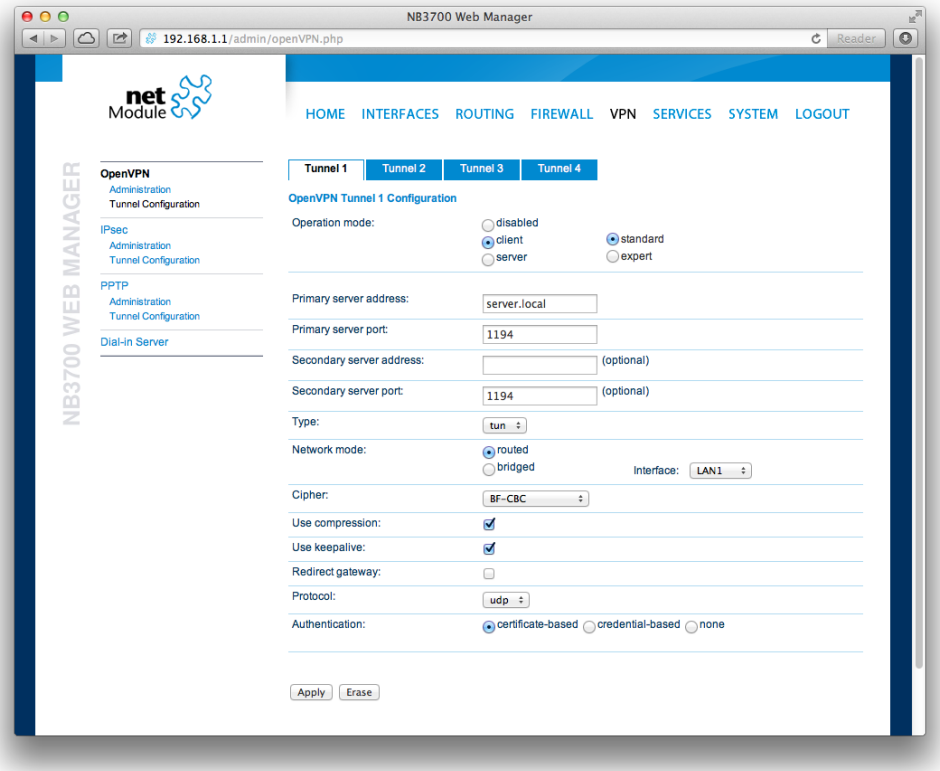


Figure 5.25.: OpenVPN Configuration

ca.crt: Root certificate authority file

server.crt: Certificate file

server.key: Private key file

dh1024.pem: Diffie-Hellman parameters file

ccd: A directory containing client-specific configuration files

Keep in mind that a certificate becomes valid once its validity time has been reached, thus an accurate system has to be set prior to creating certificates and establishing a tunnel connection. Please ensure that all NTP servers are reachable. Using host names also requires a working DNS server.

Client Management

Once you have successfully set up an OpenVPN server tunnel, you can manage and enable clients connecting to your service. Currently connected clients can be seen on this page, including the connect time and IP address. You may kick connected clients by disabling them.

In the Networking section you can specify a fixed tunnel endpoint address for each client as well as the network behind it. This can be used for routing purposes.

You may also define further routes to be pushed to each client in case you want to redirect traffic for particular networks towards the server. Routing between the clients is generally not allowed but you can enable it if desired.

Finally, you can generate and download all expert mode files for enabled clients which can be used to easily populate each client.

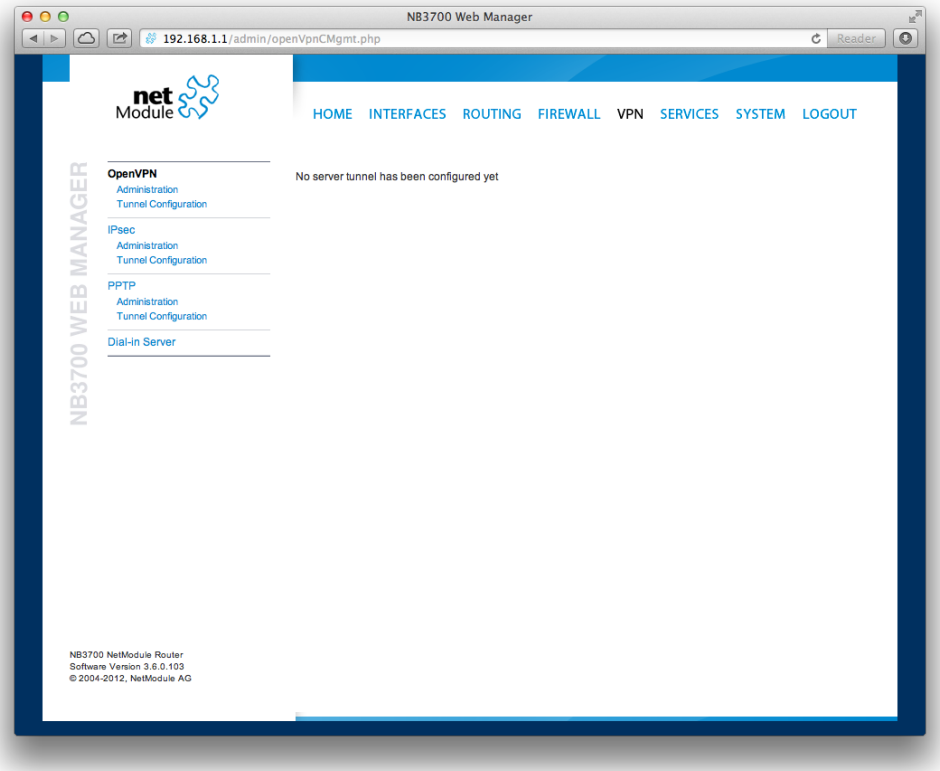


Figure 5.26.: OpenVPN Client Management

5.6.2. IPsec

IPsec is a protocol suite for securing IP communications by authenticating and encrypting each packet of a communication session and thus establishing a secure virtual private network.

IPsec includes various cryptographic protocols and ciphers for key exchange and data encryption and can be seen as one of the strongest VPN technologies in terms of security. It uses the following mechanisms:

Mechanism	Description
AH	Authentication Headers (AH) provide connectionless integrity and data origin authentication for IP datagrams and ensure protection against replay attacks.
ESP	Encapsulating Security Payloads (ESP) provide confidentiality, data-origin authentication, connectionless integrity, an anti-replay service and limited traffic-flow confidentiality.
SA	Security Associations (SA) provide a secure channel and a bundle of algorithms that provide the parameters necessary to operate the AH and/or ESP operations. The Internet Security Association Key Management Protocol (ISAKMP) provides a framework for authenticated key exchange.

Negotating keys for encryption and authentication is generally done by the Internet Key Exchange protocol (IKE) which consists of two phases:

Phase	Description
IKE phase 1	IKE authenticates the peer during this phase for setting up an ISAKMP secure association. This can be carried out by either using `main` or `aggressive` mode. The `main` mode approach utilizes the Diffie-Hellman key exchange and authentication is always encrypted with the negotiated key.The `aggressive` mode just uses hashes of the pre-shared key and therefore represents a less-secure mechanism which should generally be avoided as it is prone to dictionary attacks.
IKE phase 2	IKE finally negotiates IPSec SA parameters and keys and sets up matching IPSec SAs in the peers which is required for AH/ESP later on.

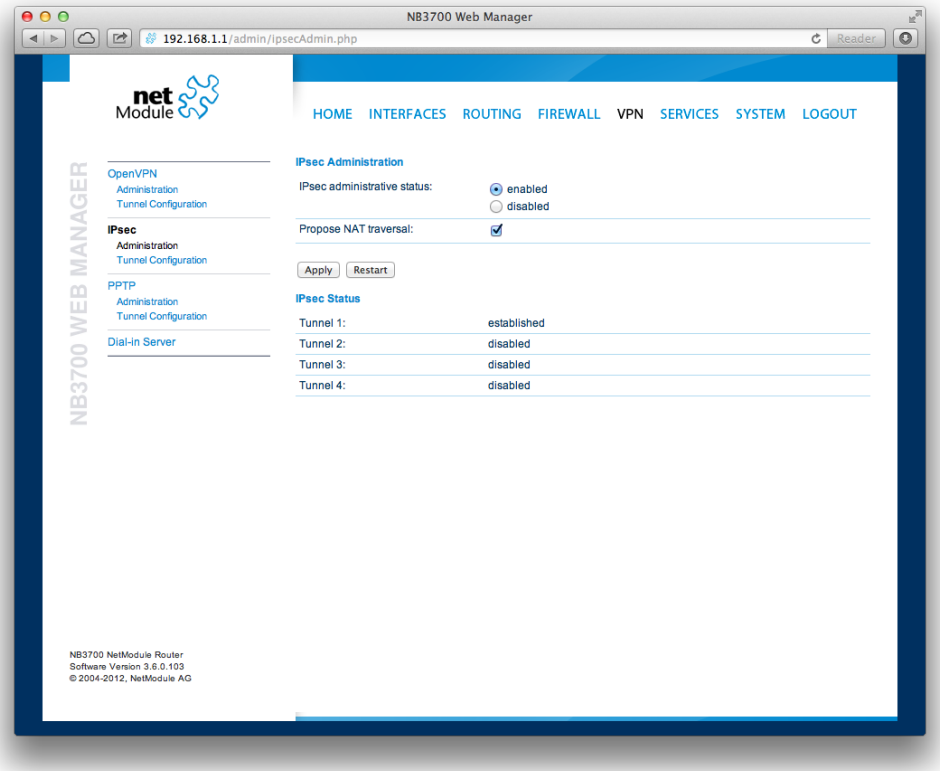


Figure 5.27.: IPsec Administration

Administration

This page can be used to enable/disable IPsec, you may also specify whether NAT-Traversal should be used.

NAT-Traversal is mainly used for connections which traverse a path where a router modifies the IP address/port of packets. It encapsulates packets in UDP and therefore requires a slight overhead which has to be taken into account when running over small-sized MTU interfaces.

Please note that running NAT-Traversal makes IKE using UDP port 4500 rather than 500 which has to be taken into account when setting up firewall rules.

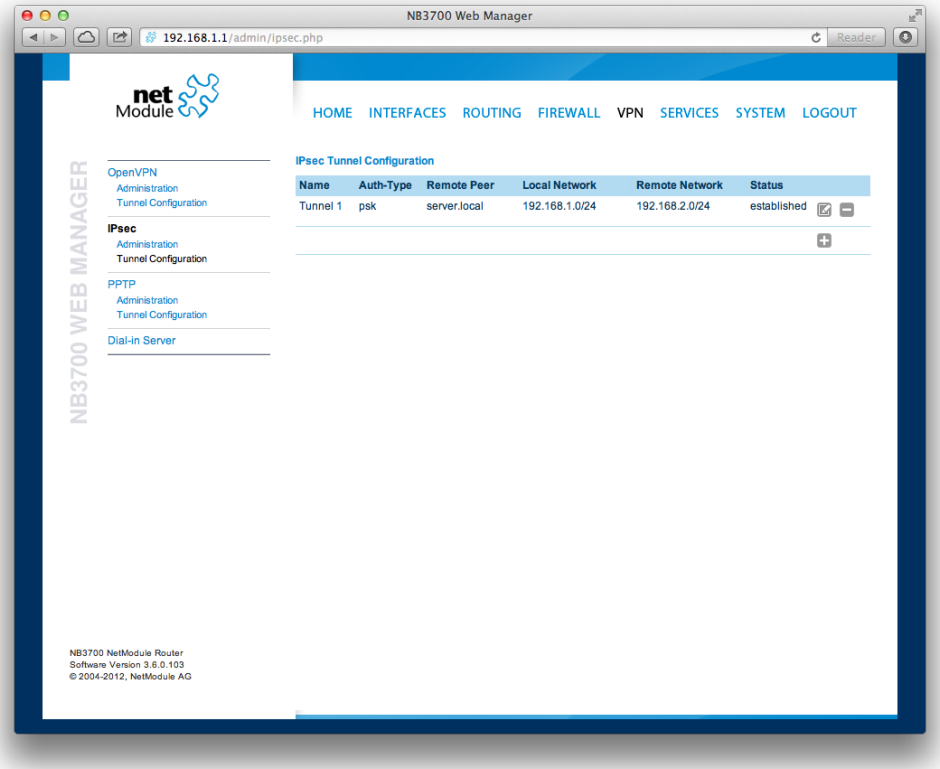


Figure 5.28.: IPsec Configuration

General

For setting up the tunnel you will have to configure the following parameters first:

Remote peer: IP address or host name of the remote IPsec peer (aka responder or server)

DPD Status: Specifies whether Dead Peer Detection (see RFC 3706) shall be used. DPD will detect any broken IPSec connections, in particular the ISAKMP tunnel, and refresh the corresponding SAs (Security Associations) and SPIs (Security

Payload Identifier) for a faster re-establishment of the tunnel.

Detection cycle): The delay (in seconds) between DPD keepalives that are sent for this connection (default 30 seconds)

Failure threshold: The number of unanswered DPD requests until the IPsec peer is considered dead (the router will then try to re-establish a dead connection automatically)

IKE Authentication

NetModule routers support IKE authentication through pre-shared keys (PSK) or certificates within a public key infrastructure.

Using PSK requires the following settings:

PSK: The pre-shared key used to authenticate at the peer

Local ID Type: The type of identification for the local ID which can be a `FQDN`, `username@FQDN` or `IP address`

Local ID: The local ID value

Local ID Type: The type of identification for the remote ID

Remote ID: The remote ID value

When using certificates you would need to specify the operation mode. When run as PKI client you can create a Certificate Signing Request (CSR) in the certificates section which needs to be submitted at your Certificate Authority and imported to the router afterwards. In PKI server mode the router represents the Certificate Authority and issues the certificates for remote peers.

IKE Proposal

This section can be used to configure the phase 1 settings:

Negotiation mode: Choose the desired negotiation mode. Preferably, `main` mode should be used but `aggressive` mode might be applicable when dealing with dynamic endpoint addresses.

Encryption algorithm: The desired IKE encryption method (we recommend AES256)

Authentication algorithm: The desired IKE authentication method (we prefer SHA1 over MD5)

IKE Diffie-Hellman Group: The IKE Diffie-Hellman Group

SA life time: The lifetime of Security Associations

Perfect Forward Secrecy: Specifies whether Perfect Forward Secrecy (PFS) should be used. This feature increases security as PFS avoids penetration of the key-exchange protocol and prevents compromisation of previous keys.

IPsec Proposal

This section can be used to configure the phase 2 settings:

Encapsulation mode: The desired encapsulation mode (Tunnel or Transport)

IPsec protocol: The desired IPsec protocol (AH or ESP)

Encryption algorithm: The desired IKE encryption method (we recommend AES256)

Authentication algorithm: The desired IKE authentication method (we prefer SHA1 over MD5)

SA life time: The lifetime of Security Associations

Networks

When creating Security Associations, IPsec will keep track of routed networks within the tunnel. Packets will be only transmitted when a valid SA with matching source and destination network is present. Therefore, you may need to specify the networks right and left of the endpoints by applying the following settings:

Local network address: The address of your local area network

Local network mask: The netmask of your local area network

Peer network address: The address of the remote network behind the peer

Peer network mask: The netmask of the remote network behind the peer

NAT address: Optionally, you can apply NAT (masquerading) for packets coming from a different local network. The NAT address must reside in the network previously specified as local network.

5.6.3. PPTP

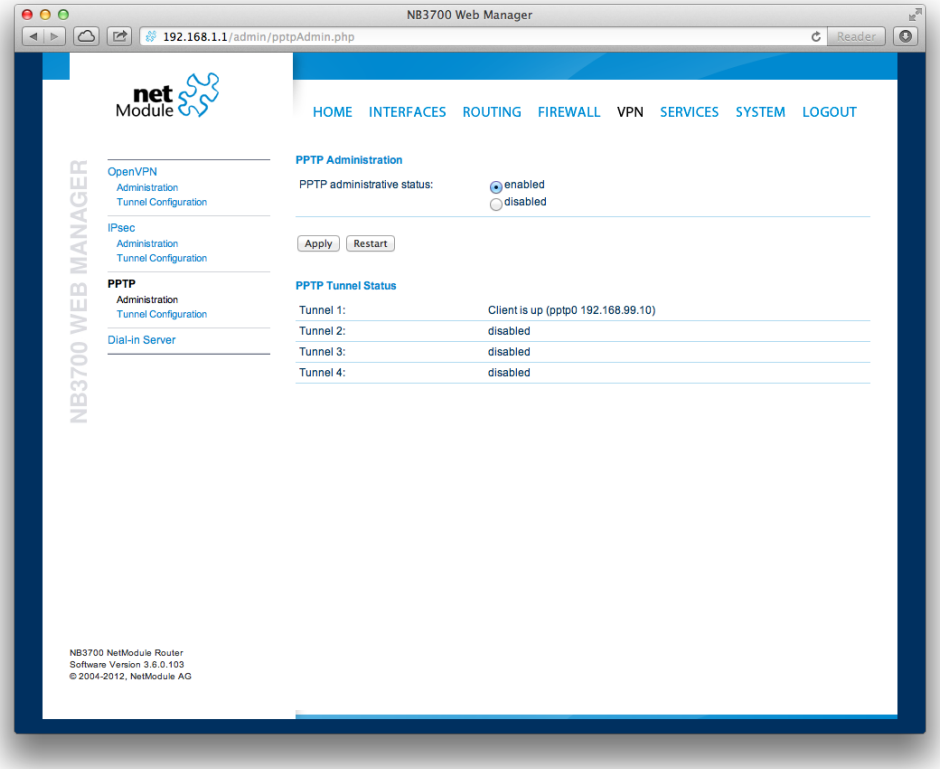


Figure 5.29.: PPTP Administration

The Point-to-Point Tunneling Protocol (PPTP) is a method for implementing virtual private networks between two hosts. PPTP is easy to configure and widely deployed amongst Microsoft Dial-up networking servers. However, due to its weak encryption algorithms, it is nowadays considered insecure but it still provides a straightforward way for establishing tunnels.

When setting up a PPTP tunnel, you would need to choose between server or client. A client tunnel requires the following parameters to be set:

Server address: The address of the remote server

Username: The user-name used for authentication

Password: The password used for authentication

Setting up a server requires the following settings:

Listen address: Specifies on which IP address should be listened for incoming client connections

Server address: The server address within the tunnel

Client address range: Specifies a range of IP addresses assigned to each client

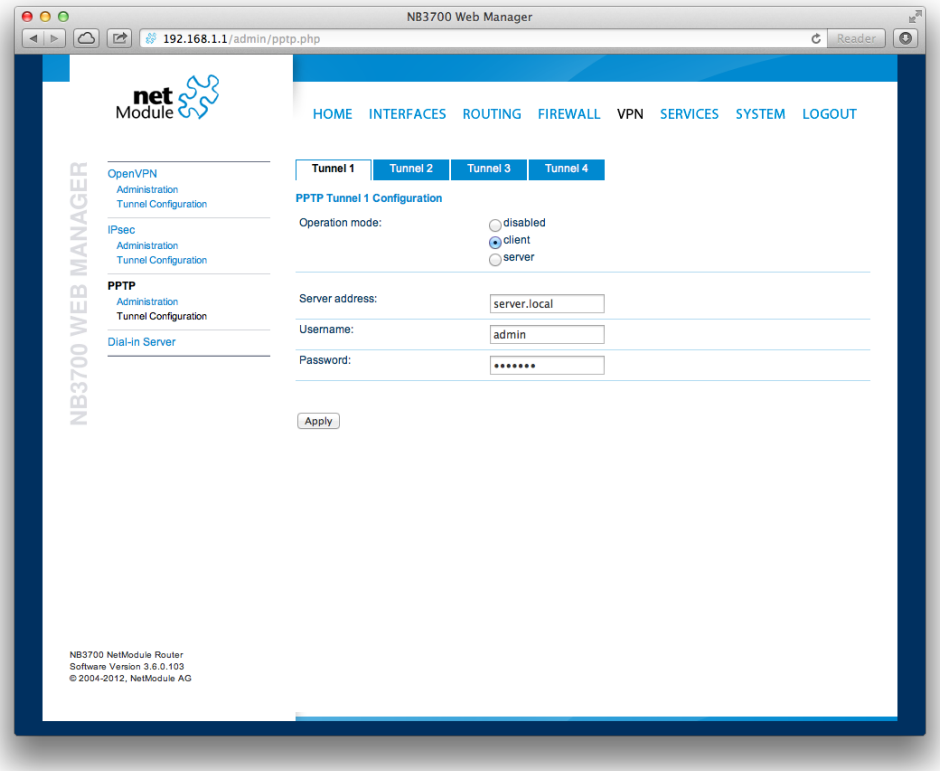


Figure 5.30.: PPTP Tunnel Configuration

PPTP Client Management

PPTP clients for a server tunnel need to be configured here. They are made up of username and password. A fixed IP address can be assigned to them which can be used to point any routes to a dedicated tunnel.

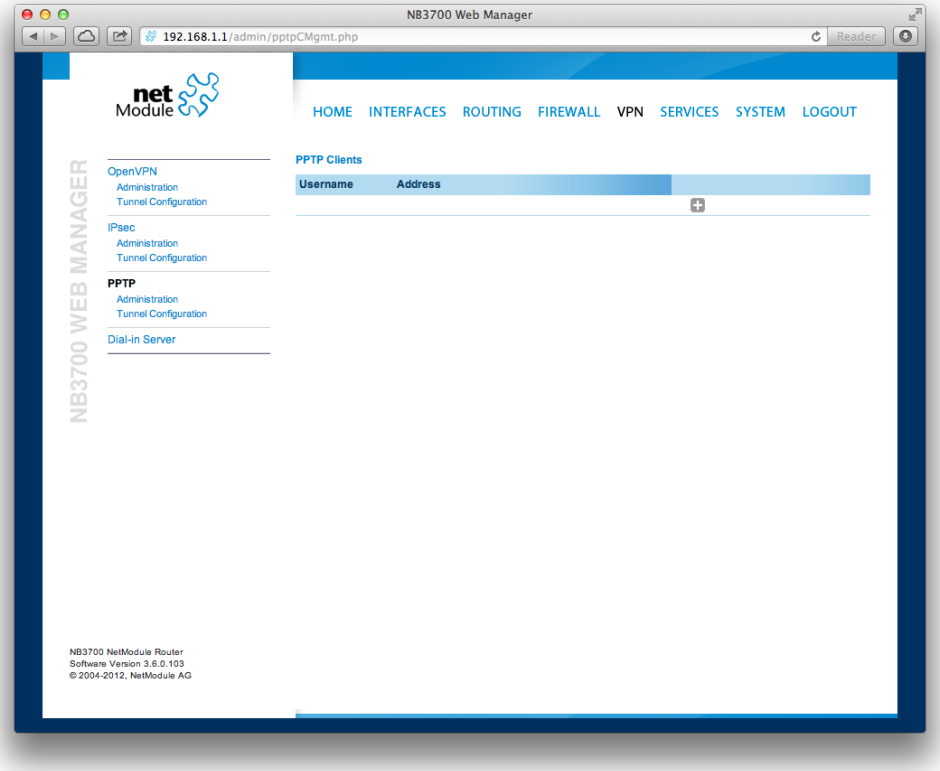


Figure 5.31.: PPTP Client Management

5.6.4. Dial-In

On this page you can configure the Dial-In server in order to establish a data connection over GSM calls. Thus, one would generally apply a required service type of 2G-only, so that the modem registers to GSM only. Naturally, a concurrent use of outgoing WWAN interfaces and Dial-In connection is not possible.

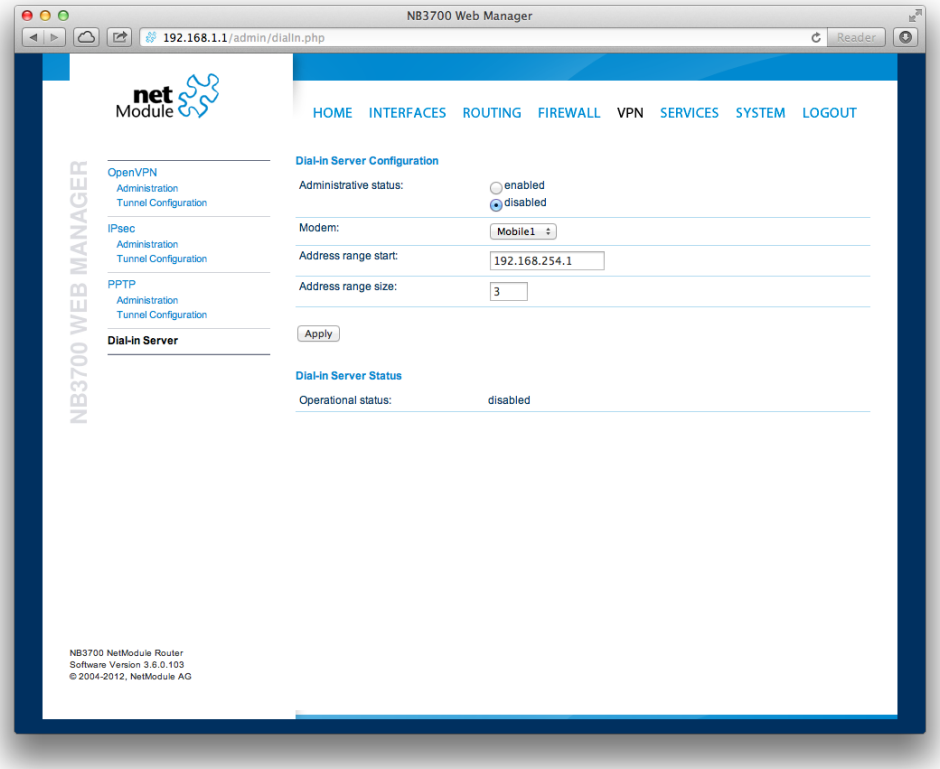


Figure 5.32.: Dial-in Server Settings

The following settings can be set:

Administrative status: Specifies whether incoming calls shall be answered or not

Modem: Specifies the modem on which calls can come in

Address range start: Start of the IP address range assigned to incoming clients

Address range size: Number of addresses for client IP address range

Besides the admin account you can configure further users in the user accounts section which shall be allowed to dial-in.

Please note that Dial-In connections are generally discouraged. As they are implemented as GSM voice calls, they suffer from unreliability and poor bandwidth.

5.7. SERVICES

5.7.1. SDK

NetModule routers are shipping with a Software Development Kit (SDK) which offers a simple and fast way to implement customer-specific functions and applications. It consists of:

1. An SDK host which defines the runtime environment (a so-called sandbox), that is, controlling access to system resources (such as memory, storage and CPU) and, by doing so, catering for the right scalability
2. An interpreter language called `arena`, a light-weight scripting language optimized for embedded systems, which uses a syntax similar to ANSI-C but adds support for exceptions, automatic memory management and runtime polymorphism on top of that
3. A NetModule-specific Application Programming Interface (API), which ships with a comprehensive set of functions for accessing hardware interfaces (e.g. digital IO ports, GPS, external storage media, serial ports) but also for retrieving system status parameters, sending E-Mail or SMS messages or simply just to configure the router

Anyone, reasonably experienced in the C language, will find an environment that is easy to dig in. However, feel free to contact us via router@support.netmodule.com and we will happily support you in finding a programming solution to your specific problem.

The Language

The `arena` scripting language offers a broad range of POSIX functions (like `printf` or `open`) and provides, together with tailor-made API functions, a simple platform for implementing any sort of applications to interconnect your favourite device or service with the router.

Here comes a short example:

```
/* We are going to eavesdrop on the first serial port
 * and turn on lights via a digital I/O output port,
 * otherwise we'd have to send a short message.
 */

for (attempts = 0; attempts < 3; attempts++) {
  if (nb_serial_read("serial0") == "Knock Knock!") {
    nb_serial_write("serial0", "Who's there?");

    if (nb_serial_read("serial0") == "Santa") {
      printf("Hurray!\n");
      nb_dio_set("out1", 1);
    }
  }
}
nb_sms_send("+123456789", "No presents this year :(")
```

A set of example scripts can be downloaded directly from the router, you can find a list of them in the appendix. The manual at http://www.netmodule.com/store/sdk gives a detailed introduction of the language, including a description of all available functions.

SDK API Functions

The current range of API functions can be used to implement the following features:

1. Send/Retrieve SMS
2. Send E-mail
3. Read/Write from/to serial device
4. Control digital input/output ports
5. Run TCP/UDP servers
6. Run IP/TCP/UDP clients
7. Access files of mounted media (e.g. an USB stick)
8. Retrieve status information from the system
9. Get or set configuration parameters
10. Write to syslog
11. Transfer files over HTTP/FTP
12. Get system events / Reboot system
13. Control the LEDs

The SDK API manual at http://www.netmodule.com/store/sdk provides an overview but also explains all functions in detail.

Please note that some functions require the corresponding services (e.g. E-Mail, SMS) to be properly configured prior to utilizing them in the SDK.

Let's now pay some attention to the very powerful API function `nb_status`. It can be used to query the router's status values in the same manner as they can be shown with the CLI. It returns a structure of variables for a specific section (a list of available sections can be obtained by running `cli status -h`).

By using the `dump` function you can figure out the content of the returned structure:

```
/* dump current location */

dump(nb_status("location"));
```

The script will then generate lines like maybe these:

```
struct(8): {
  .LOCATION_STREET       = string[11]: "Bahnhofquai"
  .LOCATION_CITY         = string[10]: "Zurich"
  .LOCATION_COUNTRY_CODE = string[2]:  "ch"
  .LOCATION_COUNTRY      = string[11]: "Switzerland"
  .LOCATION_POSTCODE     = string[4]:  "8001"
  .LOCATION_STATE        = string[6]:  "Zurich"
  .LOCATION_LATITUDE     = string[9]:  "47.3778058"
  .LOCATION_LONGITUDE    = string[8]:  "8.5412757"
}
```

In combination with the `nb_config_set` function, it is possible to start a re-configuration of any parts of the system upon status changes. You may query possible sections and parameters again with the CLI:

```
~ $ cli get -c wanlink.0
Showing configuration sections (matching 'wanlink.0'):

wanlink.0.mode
wanlink.0.name
wanlink.0.prio
wanlink.0.weight
```

Running the CLI in interactive mode, you will be also able to step through possible configuration parameters by the help of the `TAB` key.

Here is an example how one might adopt those functions:

```
/* check current city and enable the second WAN link */

location = nb_status("location");
if (location) {
    city = struct_get(location, "LOCATION_CITY");

    if (city == "Wonderland") {
        for (led = 0; led < 5; led++) {
            nb_led_set(led, LED_BLINK_FAST|LED_COLOR_RED);
        }
    } else {
        printf("You'll never walk alone in %s ...\n", city);
        nb_config_set("wanlink.1.mode=1");
    }
}
```

Running SDK

In the SDK, we are speaking of `scripts` and `triggers` which form `jobs`.

Any `arena` script can be uploaded to the router or imported by using dedicated user configuration packages. You may also edit the script directly at the Web Manager or select one of our examples. You will further have a testing section on the router which can be used to check your syntax or doing test runs.

Once uploaded, you will have to specify a trigger, that is, telling the router when the script is to be executed. This can be either time-based (e.g. each Monday) or triggered by one of the pre-defined system events (e.g. wan-up) as described in Events chapter 5.7.6. With both, a script and a trigger, you can finally set up an SDK job now. The `test` event usually serves as a good facility to check whether your job is doing well. The admin section also offers facilities to troubleshoot any issues and control running jobs.

The SDK host (sdkhost) corresponds to the daemon managing the scripts and their operations and thus avoiding any harm to the system. In terms of resources, it will limit CPU and memory for running scripts and also provide a pre-defined portion of the available flash storage. You may, however, extend it by external USB storage or (depending on your model) SD cards.

Files written to `/tmp` will be hold in memory and will be cleared upon a restart of the script. As your scripts operate in the sandbox, you will have no access to tools on the system (such as `ifconfig`).

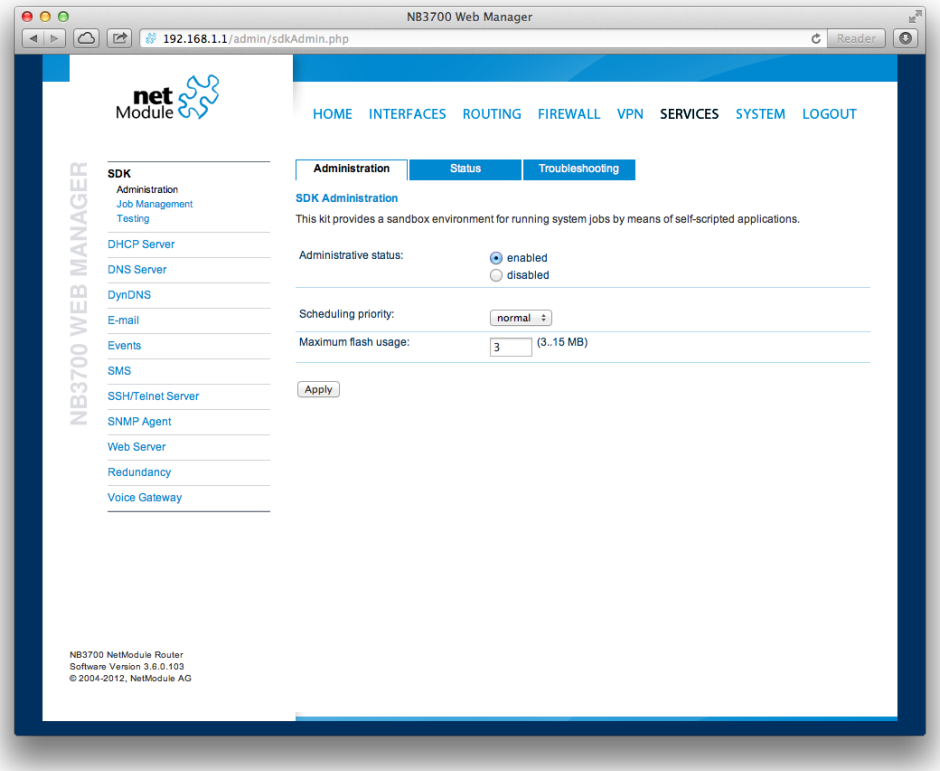


Figure 5.33.: SDK Administration

Administration

This page can be used to control the SDK host and apply the following settings:

Parameter: Description

Administrative status: Specifies whether SDK scripts should run or not

Scheduling priority: Specifies the process priority of the sdkhost, higher priorities will speed up scheduling your scripts, lower ones will have less impact to the host system

Maximum flash usage: The maximum amount of MBytes your scripts can write to the internal flash

The status page informs you about the current status of the SDK. It provides an overview about any finished jobs, you can also stop a running job there and view the script output in the troubleshooting section where you will also find links for downloading the manuals and examples.

Job Management

This page can be used to set up scripts, triggers and jobs. It is usually a good idea to create a trigger first which is made up by the following parameters:

Name: A meaningful name to identify the trigger

Type: The type of the trigger, either time-based or event-based

Condition: Specifies the time condition for time-based triggers (e.g. hourly)

Timespec: The time specification which, together with the condition, specifies the time(s) when the trigger should be pulled

Event: The system event upon which the trigger should be pulled

You can now add your personal script to the system by applying the following parameters:

Name: A meaningful name to identify the script

Description: An optional description of the script

Arguments: An optional set of arguments passed to the script (supports quoting)

Action: You may either edit a script, upload it to the system or select one of the example scripts or an already uploaded script

You are ready to set up a job afterwards, it can be created by using the following parameters:

Name: A meaningful name to identify the job

Trigger: Specifies the trigger that should launch the job

Script: Specifies the script to be executed

Arguments: Defines arguments which can be passed to the script (supports quoting), they will precede the arguments you formerly may have assigned to the script itself

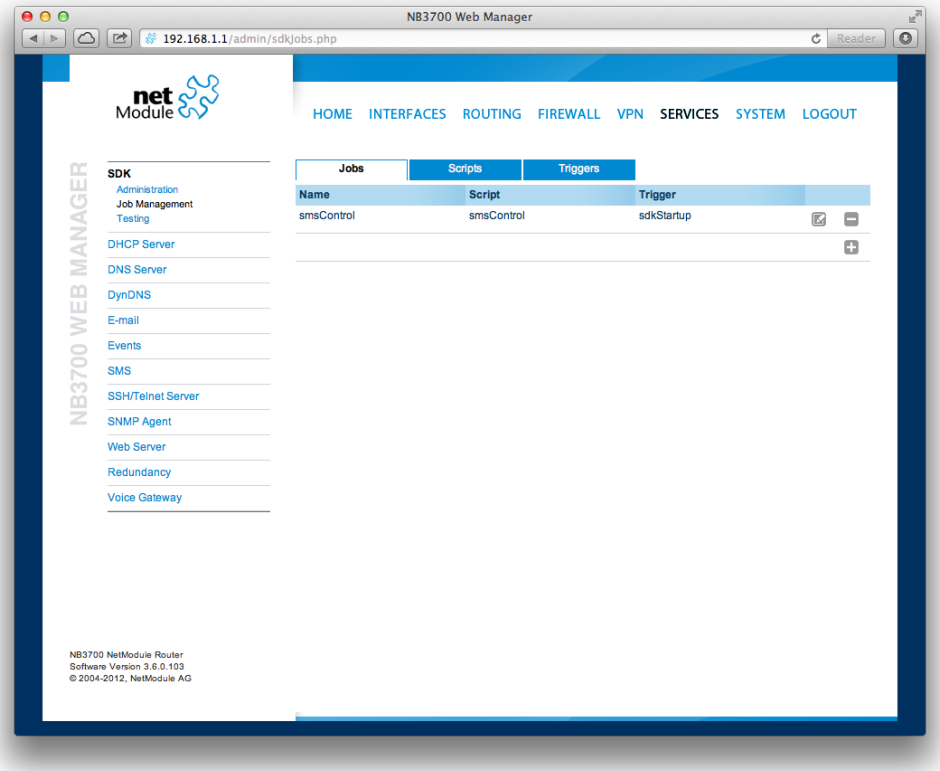


Figure 5.34.: SDK Jobs

Testing

The testing page offers an editor and an input field for optional arguments which can be used to perform test runs of your script or test dedicated portions of it. Please note that you might need to quote arguments as they will otherwise be separated by white-spaces.

```
/* arguments: 'schnick schnack "s c h n u c k"'

for (i = 0; i < argc; i++) {
    printf("argv%d: %s\n", argv[i]);
}

/* generates:
 *      argv0: scriptname
 *      argv1: schnick
 *      argv2: schnack
 *      argv3: s c h n u c k
 */
```

In case of syntax errors, `arena` will usually print error messages as follows (indicating the line and position where the parsing error occurred):

```
/scripts/testrun:2:10:FATAL: parse error, unexpected $, expecting ';'
```

SDK Sample Application

As an introduction, you can step through a sample application, namely the SMS control script, which implements remote control over short messages and can be used to send a status of the system back to the sender. The source code is listed in the appendix.

Once enabled, you can send a message to the phone number associated with a SIM / modem. It generally requires a password to be given on the first line and a command on the second, such as:

```
admin01
status
```

We strongly recommend to use authentication in order to avoid any unintended access, however you may pass `noauth` as argument to disable it. You can then skip the first line containing the password. Having a closer look to the script, you will see that you will also be able to restrict the list of permitted senders. Please inspect the system log for troubleshooting any issues.

The following commands are supported:

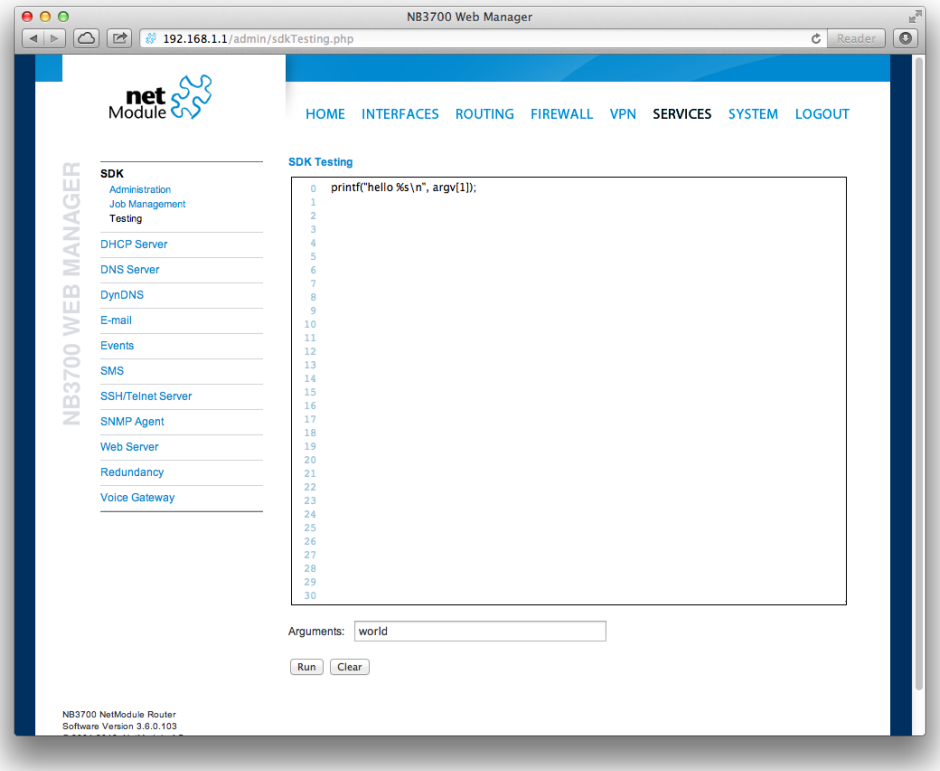


Figure 5.35.: SDK Testing

Command	Action
status	Will reply a message to the sender including a short system overview
connect	Will enable the first WAN link configured on the system
disconnect	Will disable the first WAN link configured on the system
reboot	Initiates a reboot of the system
output 1 on	Turns on the first digital output port
output 1 off	Turns off the first digital output port
output 2 on	Turns on the second digital output port
output 2 off	Turns off the second digital output port

Table 5.5.: SMS Control Commands

A response to the status command typically looks like:

```
System: NB2700 hostname (00:11:22:AA:BB:CC)
WAN1: WWAN1 is up (10.0.0.1, Mobile1, UMTS, -83 dBm, LAI 12345)
GPS: lat 47.377894, lon 8.540055, alt 282.200
OVPN: client on tun0 is up (10.0.8.4)
DIO: IN1=off, IN2=off, OUT1=on, OUT2=off
```

5.7.2. DHCP Server

This section can be used to individually configure the Dynamic Host Configuration Protocol (DHCP) service for each LAN interface which will serve dynamic IP addresses to hosts in the local network. You may also have a look to the leases page where you can find an overview about negotiated client addresses.

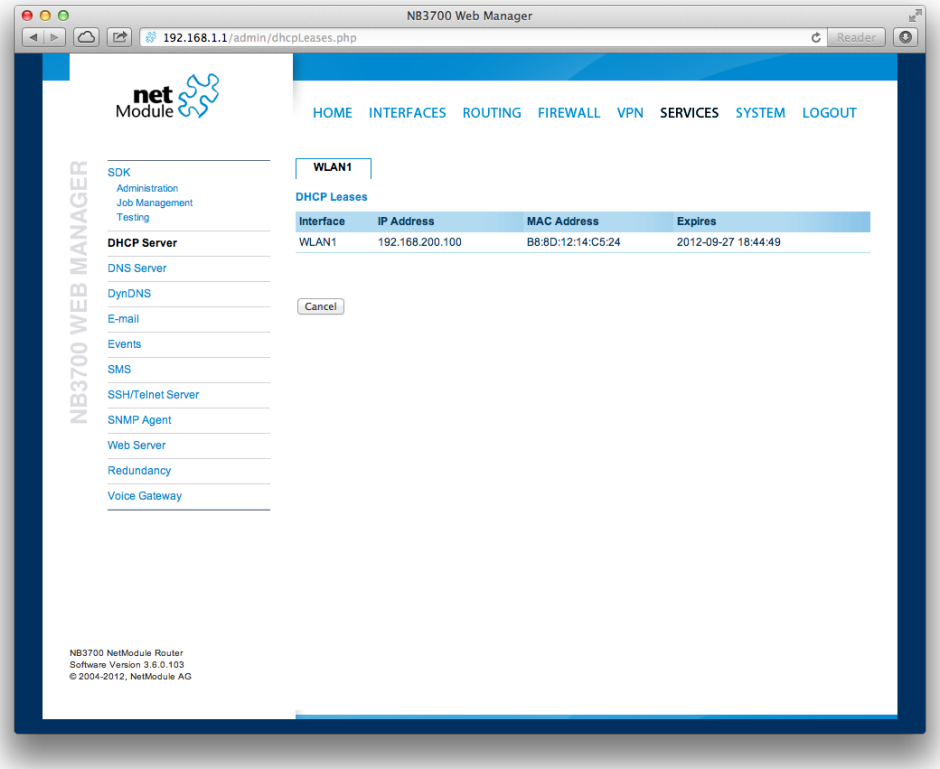


Figure 5.36.: DHCP Leases

Please note that WLAN interfaces (for each SSID) will pop up here as well in case you have configured an access point respectively.
The following settings for each interface can be applied then:

Administrative status: Specifies whether the DHCP server is enabled or not

First lease address: The first address out of the range of IP addresses given to hosts

Last lease address: The last address out of this range

Lease duration: Number of seconds how long a given lease shall be valid until it has to be requested again

Persistent leases: By turning on this option the router will remember issued leases even after a reboot. This can be used to ensure that the same IP address will be assigned to a particular host.

DHCP options: By default the DHCP will hand out the interface address as default gateway and the current DNS server addresses if not configured elsewise. You can specify fixed addresses here.

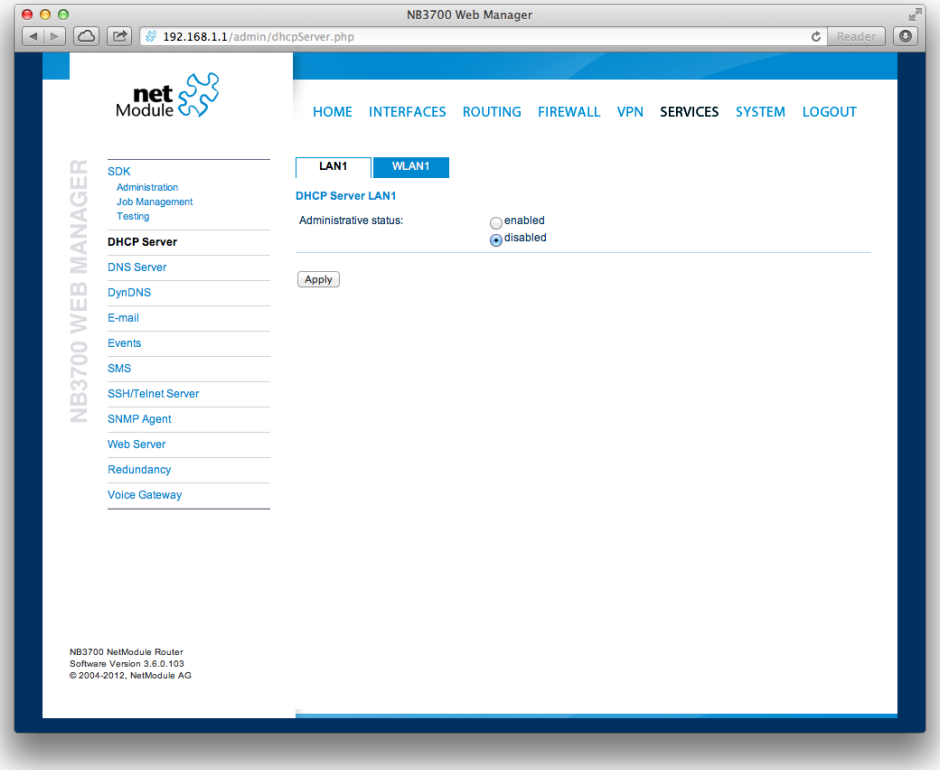


Figure 5.37.: DHCP Server

5.7.3. DNS Server

The DNS server can be used to proxy DNS requests towards servers on the net which have for instance been negotiated during WAN link negotiation. By pointing DNS requests to the router, one can reduce outbound DNS traffic as it is caching already resolved names but it can be also used for serving fixed addresses for particular host names.

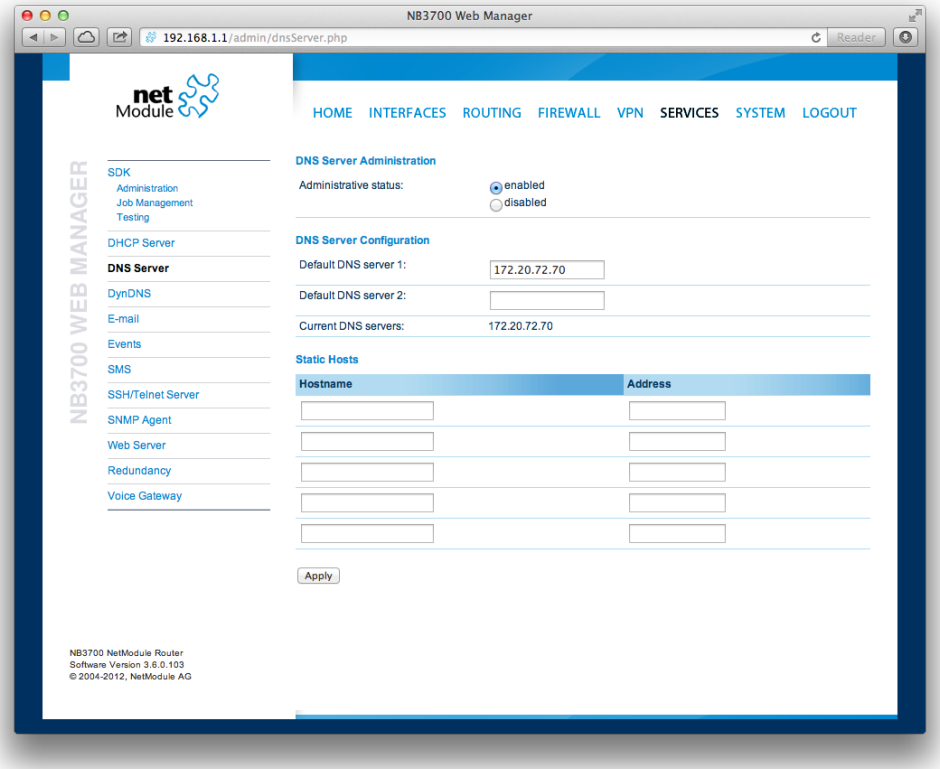


Figure 5.38.: DNS Server

The following settings can be applied:

Administrative status: Enables or disables the DNS server

Default DNS server 1: The primary default DNS server which will be used if no other service can be negotiated

Default DNS server 2: The secondary server which will be used in case the primary server is not available

You may further configure static hosts for serving fixed IP addresses for various hostnames. Please remember to point local hosts to the router's address for resolving them.

5.7.4. DynDNS

The dynamic DNS client on this box can be used to tell one or more DynDNS providers the current WAN address of this system. This address can be either derived from the current hot-link address or by querying an HTTP service in the Internet for the current Internet address. The latter might be applicable in NAT scenarios.

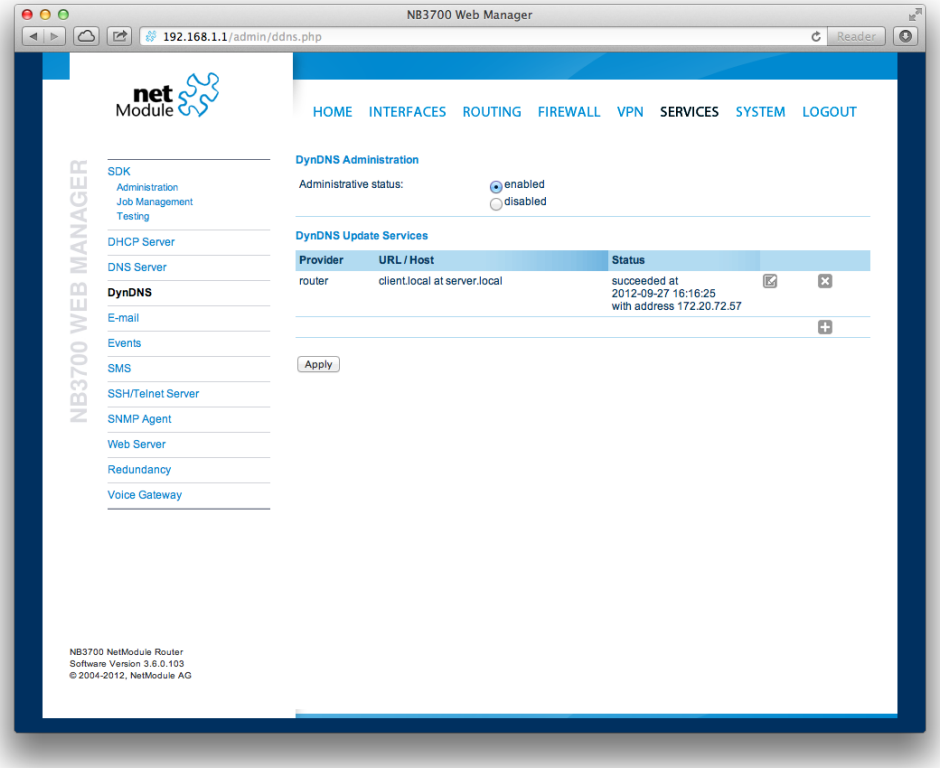


Figure 5.39.: Dynamic DNS Settings

Each service can be configured as follows:

Provider: You can choose one of the listed providers or provide a custom URL

Dynamic address: Specifies whether the address is derived from the hot-link or via an external service

Hostname: The host-name provided by your DynDNS service (e.g. mybox.dyndns.org)

Port: The HTTP port of the service (typically 80)

Username: The user-name used for authenticating at the service

Password: The password used for authentication

Please note that your NetModule router can operate as DynDNS service as well, provided that you hold a valid SERVER license and have your hosts pointed to the DNS service of the router.

5.7.5. E-Mail

The E-Mail client can be used to send notifications to a particular E-Mail address upon certain events or by SDK scripts.

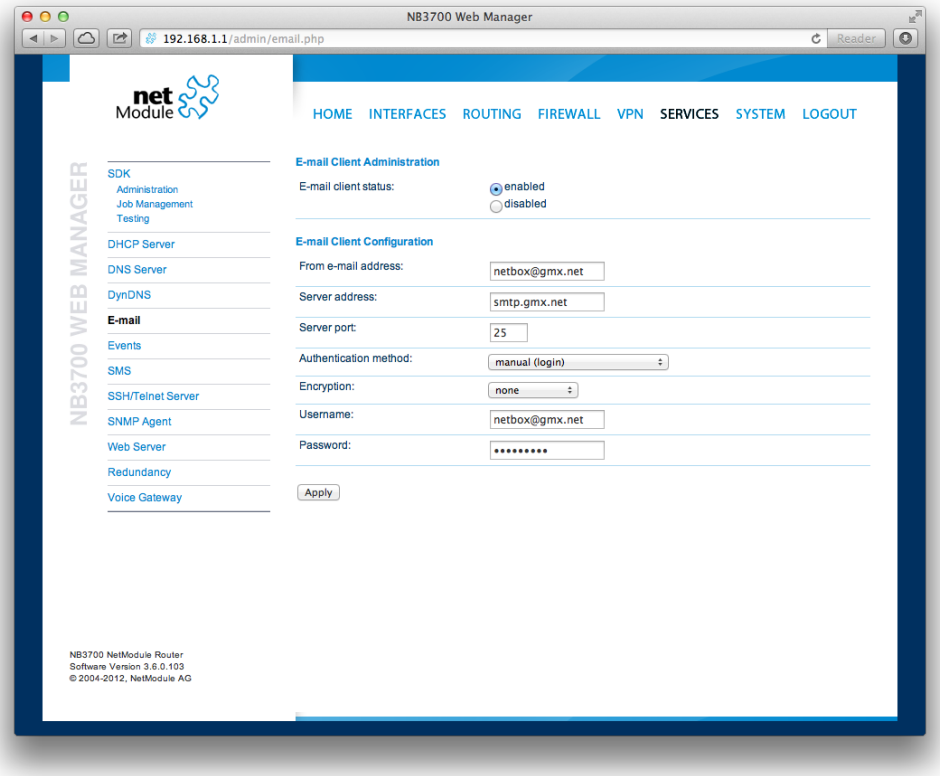


Figure 5.40.: E-Mail Settings

It can be enabled by applying the following settings.

E-mail client status: Administrative status of the E-Mail client

From e-mail address: E-Mail address of the sender

Server address: SMTP server address

Server port: SMTP server port (typically 25)

Authentication method: Select the required authentication method which will be used to authenticate against the SMTP server

Username: User name used for authentication

Password: Password used for authentication

5.7.6. Events

By using the event manager you can notify one or more recipients by SMS or E-Mail upon certain system events. The messages will contain a description provided by you and a short system info.

A list of all system events can be found in the appendix A.2.

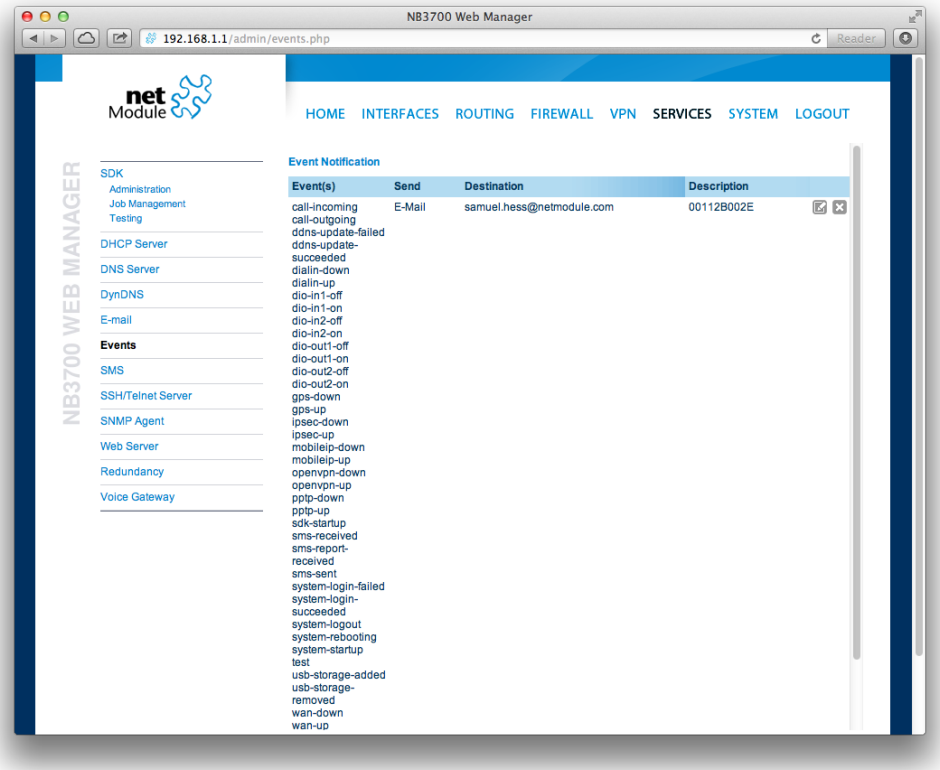


Figure 5.41.: Event Notification Settings

5.7.7. SMS

Administration

On NetModule routers it is possible to receive or send short messages (SMS) over each mounted modem (depending on the assembly options). Messages are received by querying the SIM card over a modem, so prior to that, the required assignment of a SIM card to a modem needs to be specified on the SIMs page.

Please bear in mind, in case you are running multiple WWAN interfaces sharing the same SIM, that the system may switch SIMs during operation which will also result in different settings for SMS communication.

Received messages are pulled from the SIMs and temporarily stored on the router but get cleared after a system reboot. Please consider to consult an SDK script in case you want to process or copy them.

Sending messages heavily depends on the registration state of the modem and whether the provided SMS Center service works and may fail. You may use the `sms-report-received` event to figure out whether a message has been successfully sent.

Please do not forget that modems might register roaming to foreign networks where other fees may apply. You can manually assign a fixed network (by LAI) in the SIMs section.

The relevant page can be used to enable the SMS service and specify on which it should operate.

Routing & Filtering

By using SMS routing you can specify outbound rules which will be applied whenever message are sent. On the one hand, you can forward them to an enabled modem. For a particular number, you can for instance enforce messages being sent over a dedicated SIM. Phone numbers can also be specified by regular expressions, here are some examples:

Number	Result
+12345678	Specifies a fixed number
+1*	Specifies any numbers starting with `+1`
+1*9	Specifies any numbers starting with `+1` and ending with `9`
+[12]*	Specifies any numbers starting with either `+1` or `2`

Table 5.6.: SMS Number Expressions

Please note that numbers have to be entered in international format including a valid

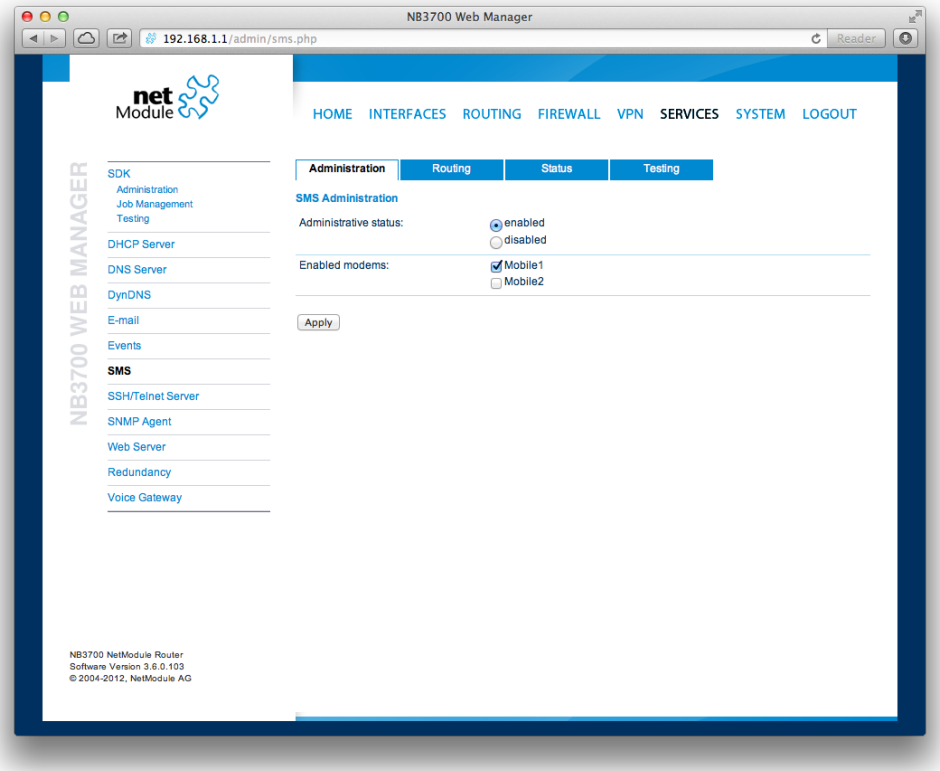


Figure 5.42.: SMS Configuration

prefix.

On the other hand, you can also define rules to drop outgoing messages, for instance, when you want to avoid using any expensive service or international numbers.

Both types of rules form a list will be processed by order, forwarding outgoing messages over the specified modem or dropping them. Messages which are not matching any of the rules below will be dispatched to the first available modem.

Filtering serves a concept of firewalling incoming messages, thus either dropping or allowing them on a per-modem basis. The created rules are processed by order and in case of matches will either drop or forward the incoming message before entering the system. All non-matching messages will be allowed.

Status

The status page can be used to the current modem status and get information about any sent or received messages. There is a small SMS inbox reader which can be used to view or delete the messages. Please note that the inbox will be cleared each midnight in case it exceeds 512 kBytes of flash usage.

Testing

This page can be used to test whether SMS sending in general or filtering/routing rules works. The maximum length per message part is limited to 160 characters, we also suggest to exclusively use characters which are supported by the GSM 7-bit alphabet.

5.7.8. SSH/Telnet Server

Apart from the Web Manager, the SSH and Telnet services can be used to log into the system. Valid users include *root* and *admin* as well as additional users as they can be created in the User Accounts section. Please note, that a regular system shell will only be provided for the *root* user, the CLI will be launched for any other user whereas normal users will only be able to view status values, the *admin* user will obtain privileges to modify the system.

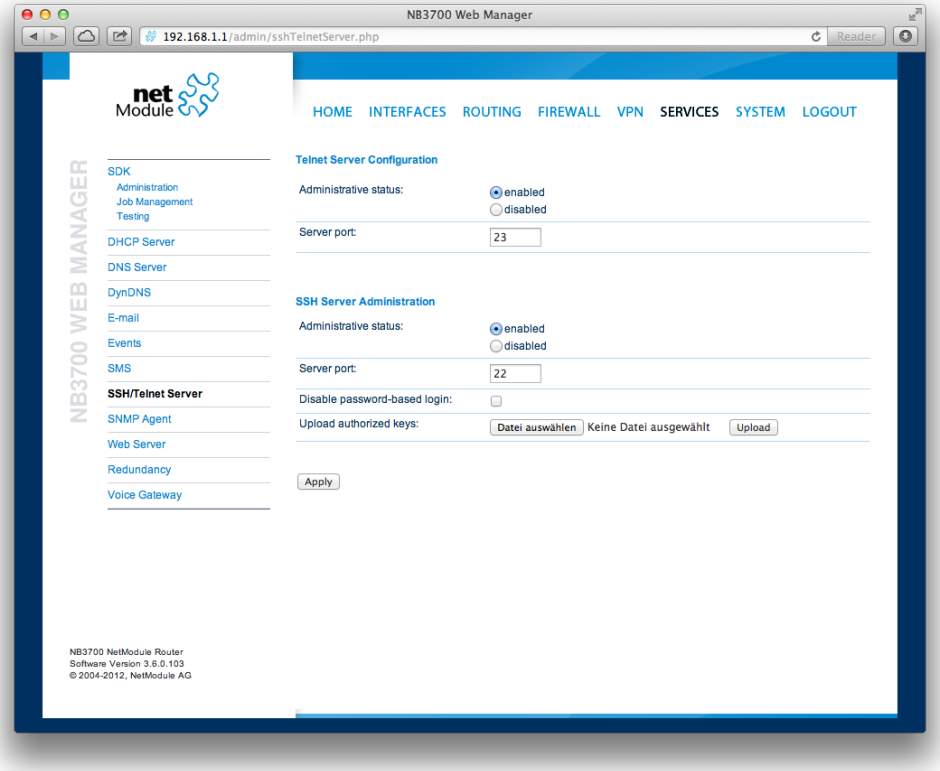


Figure 5.43.: SSH and Telnet Server

Please note that these services will be accessible from the WAN interface also. In doubt, please consider to disable or restrict access to them by applying applicable firewall rules.
The following parameters can be applied to the Telnet service:

Administrative status: Whether the Telnet service is enabled or disabled
Server port: The TCP port of the service (usually 23)

The following parameters can be applied to the SSH service:

Administrative status: Whether the SSH service is enabled or disabled
Server port: The TCP port of the service (usually 22)

Disable password-based login: By turning on this option, all users will have to authenticate by SSH keys which can be uploaded to the router.

5.7.9. SNMP Agent

NetModule routers are equipped with an SNMP daemon, supporting basic MIB tables (such as ifTable), plus additional enterprise MIBs to manage multiple systems. The corresponding VENDOR MIB can be found in the appendix or downloaded from the router.

Setting MIB values is limited to SNMPv3 and only the admin user is entitled to trigger the extensions. They offer facilities for:

- rebooting the device
- updating to a new system software via FTP/TFTP/HTTP
- updating to a new system configuration via FTP/TFTP/HTTP
- getting WWAN/GNSS/WLAN/DIO information

The SNMP extensions can be read and triggered as follows:

Listing 5.1: Getting the software version of the system:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.1.0
```

Listing 5.2: Getting the kernel version:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.2.0
```

Listing 5.3: Getting the serial number:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.3.0
```

Listing 5.4: Restarting the device:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.10.0 i 1
```

Listing 5.5: Running a configuration update:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.11.0 s
"http://server/directory"
```

You can use TFTP, HTTP, HTTPS and FTP URLs, specifying a username/password or a port is not yet supported. Please note that config updates expect a zip-file named `<serial-number>.zip` in the specified directory.

Listing 5.6: Getting the configuration update status:

```
snmpget -v 3 -u snmpadmin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   snmpadmin 192.168.1.1  1.3.6.1.4.1.31496.10.40.12.0
```

The return value can be one of: `succeeded (1)`, `failed (2)`, `inprogress (3)`, `notstarted (4)`.

Listing 5.7: Running a software update:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.13.0 s
"http://server/directory"
```

Listing 5.8: Getting the software update status:

```
snmpget -v 3 -u snmpadmin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   snmpadmin 192.168.1.1  1.3.6.1.4.1.31496.10.40.14.0
```

The return value can be one of: `succeeded (1)`, `failed (2)`, `inprogress (3)`, `notstarted (4)`.

The following parameters can be used to set up the SNMP agent:

SNMP agent status: Enable or disable the SNMP agent

Listening Port: SNMP agent port

Community: An SNMP community string corresponding to the group that devices and management stations running SNMP belong to

Contact: System maintainer/contact information

Location: Location of the device

Trap target host: The host where the traps will be sent to

Trap target port: The port where the traps will be sent to

Signal strength trap threshold: A trap will be sent, if signal strength falls below this threshold.

Signal strength trap reactivation threshold: No further traps will be sent as long as signal strength his not higher than this value.

Attention must be paid to the fact that SNMP passwords have to be more than 8 characters long. Shorter passwords will be doubled for SNMP (e.g. `admin01` becomes `admin01admin01`).

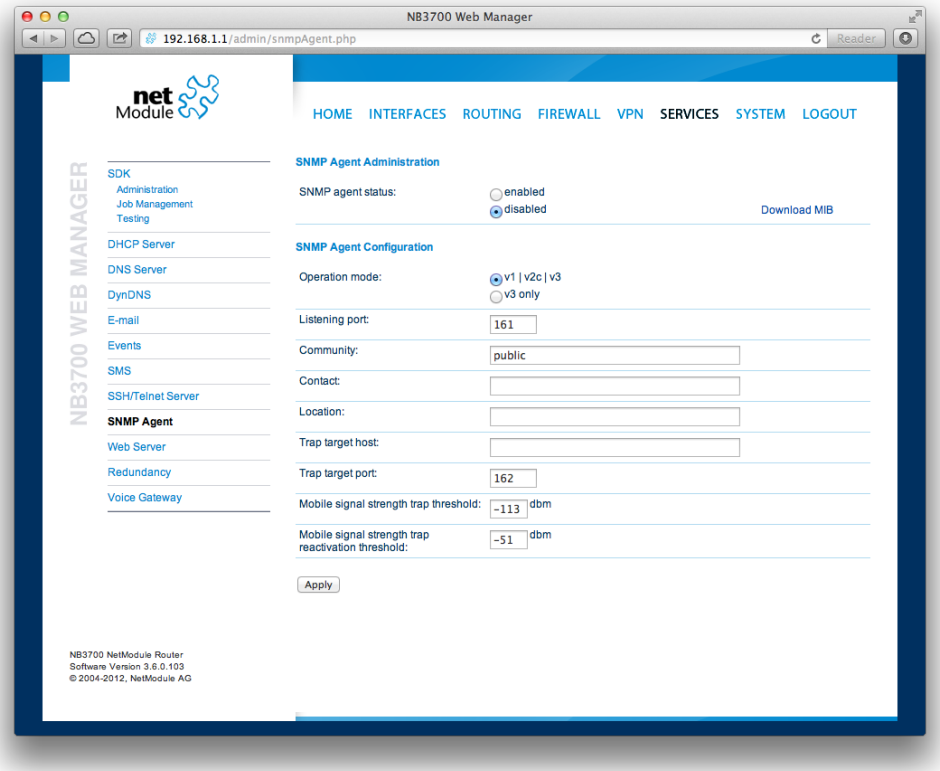


Figure 5.44.: SNMP Agent

Once the SNMP agent is enabled, SNMP traps are generated in case of the following events:

- Start-up of the box
- Shutdown of the box
- VPN connected
- VPN disconnected
- Signal strength fell below the strength trap threshold

The start-up trap is implemented using the standard cold start & warm start traps. The system-shutdown trap is sent, when the system is rebooted via the reboot function of the web interface or when the watchdog reboots the system.

5.7.10. Web Server

This page can be used to configure different ports for accessing the Web Manager via HTTP/HTTPS. We strongly recommend to use HTTPS when accessing the web service via a WAN interface as the communication will be encrypted and thus avoids any misuse of the system.

In order to enable HTTPS you would need to generate or upload a server certificate in the section 5.8.6.

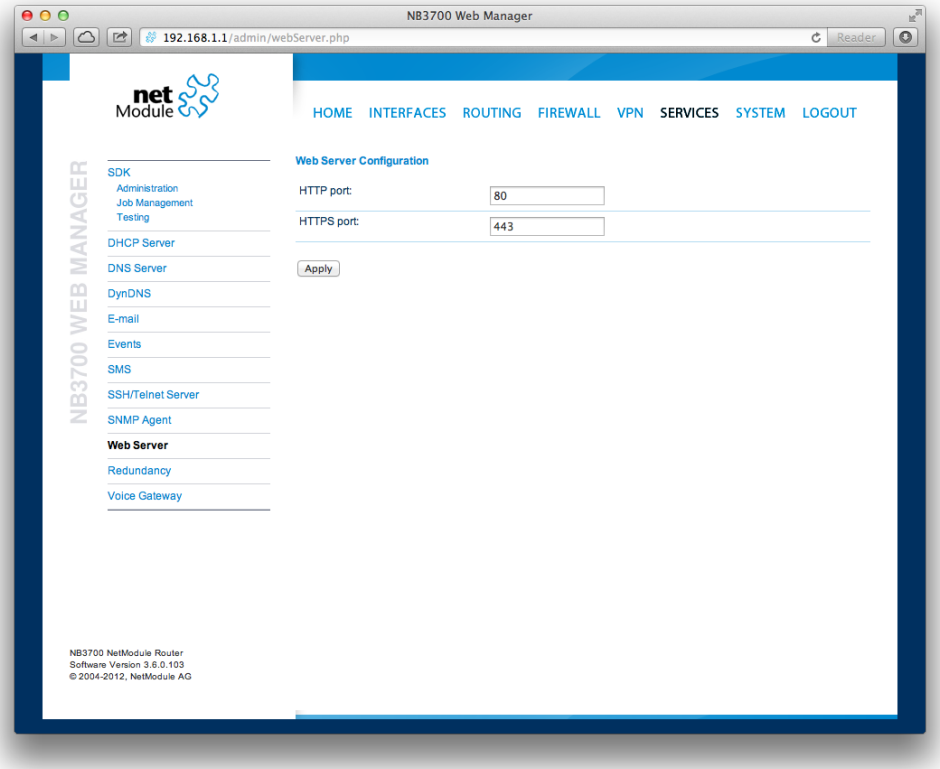


Figure 5.45.: Web Server

HTTP port: Web server port for HTTP connections
HTTPS port: Web server port for HTTPS connections

5.7.11. Redundancy

This page can be used to set up a redundant pair of NetModule routers (or other systems) by running the Virtual Router Redundancy Protocol (VRRP) between them. A typical VRRP scenario defines a first host playing the master and another the backup device, they both define a virtual gateway IP address which will be distributed by gratuitous ARP messages for updating the ARP cache of all LAN hosts and thus redirecting the packets accordingly. A takeover will happen within approximately 3 seconds as soon as the partner is not reachable anymore (checked via multicast packets). This may happen when one device is rebooting or the Ethernet link went down. Same applies when the WAN link goes down.

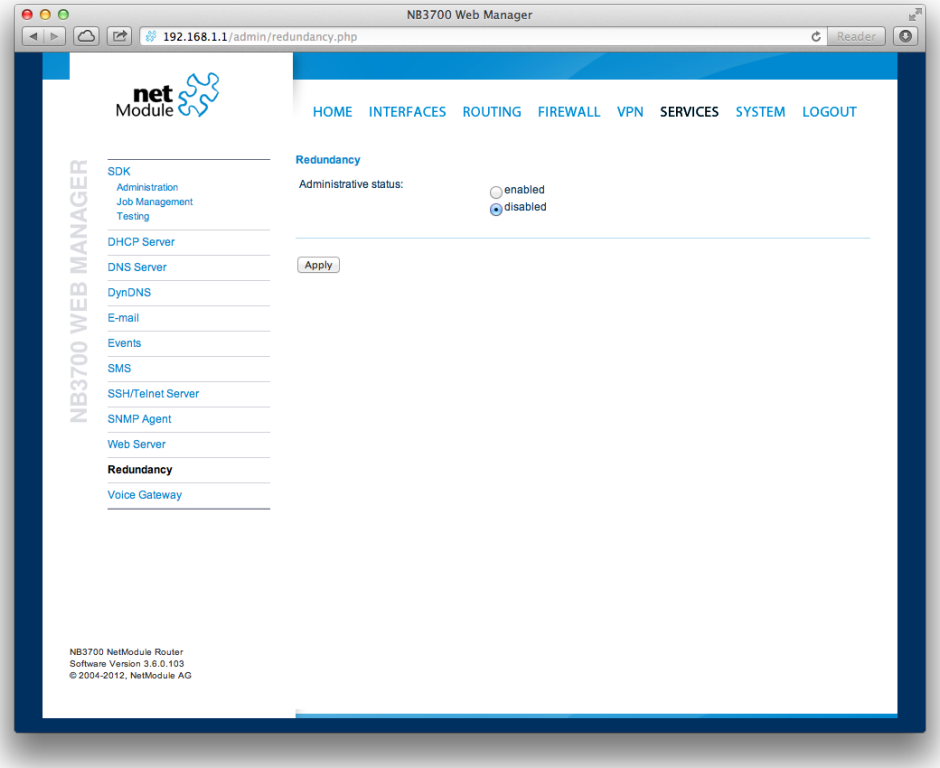


Figure 5.46.: VRRP Configuration

In case DHCP has been activated, please keep in mind that you will need to reconfigure the DHCP gateway address offered by the server and let them point to the virtual gateway address. In order to avoid conflicts you may turn off DHCP on the backup device or even better, split the DHCP lease range across both routers in order to prevent any lease duplication.

Administrative status: Administrative status

Role: The role of this system (either master or backup)

VID: The Virtual Router ID (you can theoretically run multiple instances)

Interface: Interface on which VRRP should be performed

Virtual gateway address: The virtual gateway address formed by the participating hosts

We assign a priority of `100` to the master and `1` to the backup router. Please adapt the priority of your third-party device appropriately.

5.7.12. Voice Gateway

Depending on your hardware, you can set up a voice gateway on the router which can be connected by any VoIP client from the local network capable of the SIP protocol. It hereby listens for arriving SIP calls and forwards them as a GSM call on the modem which has been configured. Due to this nature only one concurrent call is possible.

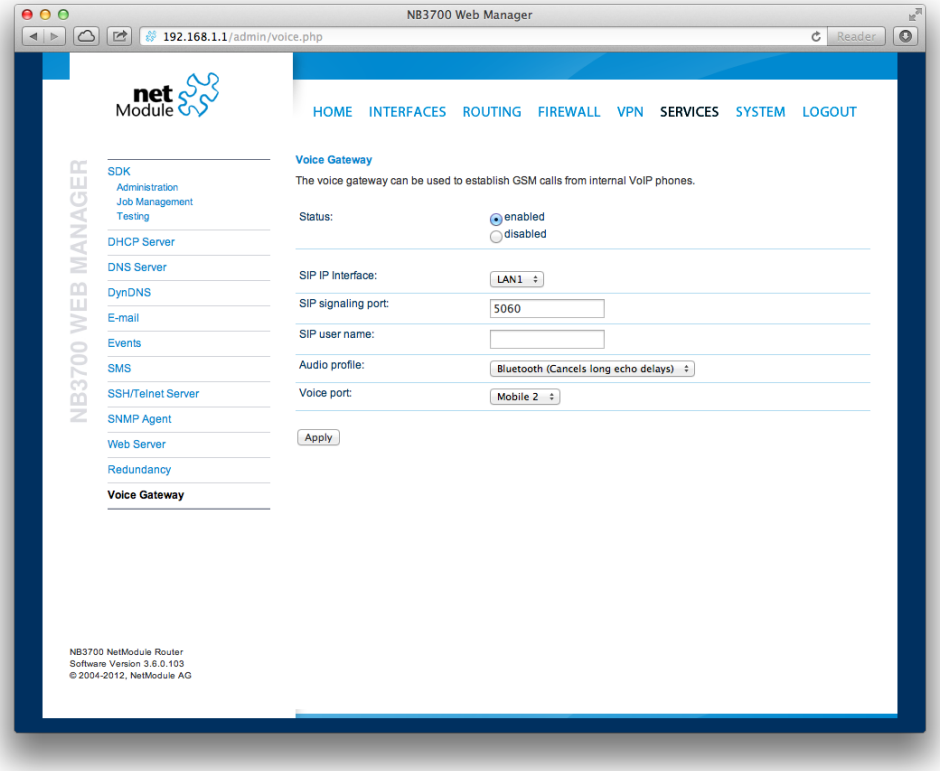


Figure 5.47.: Voice Gateway

The following parameters can be used to set it up:

Administrative status: Specifies whether the gateway shall be enabled or disabled

SIP interface: Specifies the local interface (LAN or WLAN) to which should be listened for incoming calls

SIP port: Specifies the port on which should be listened

SIP user name: reserved for future use

Audio profile: Selects the audio profile which should be applied to outgoing calls. This parmeter influences echo cancelation. For nomal use select *Bluetooth*

Voice port: Selects the modem on which GSM calls shall be established

Please bear in mind, in case you are running multiple WWAN interfaces sharing the same SIM, that the system may switch SIMs during operation which will also result in

different settings for voice communication.

Client Configuration

The sip client should be configured to use the router as a voice gateway. The easiest way to achieve this is to configure the router as proxy. The Voice Gateway does not require authenticationi however it may be necessary to fill in dummy values as user ID, Domain and Password. Any SIP client with access to the *SIP IP Interface* can use the router as a voice gateway.

Sample configuration for the Counter Path X-Lite client (Version 5.0.0 build 67284)

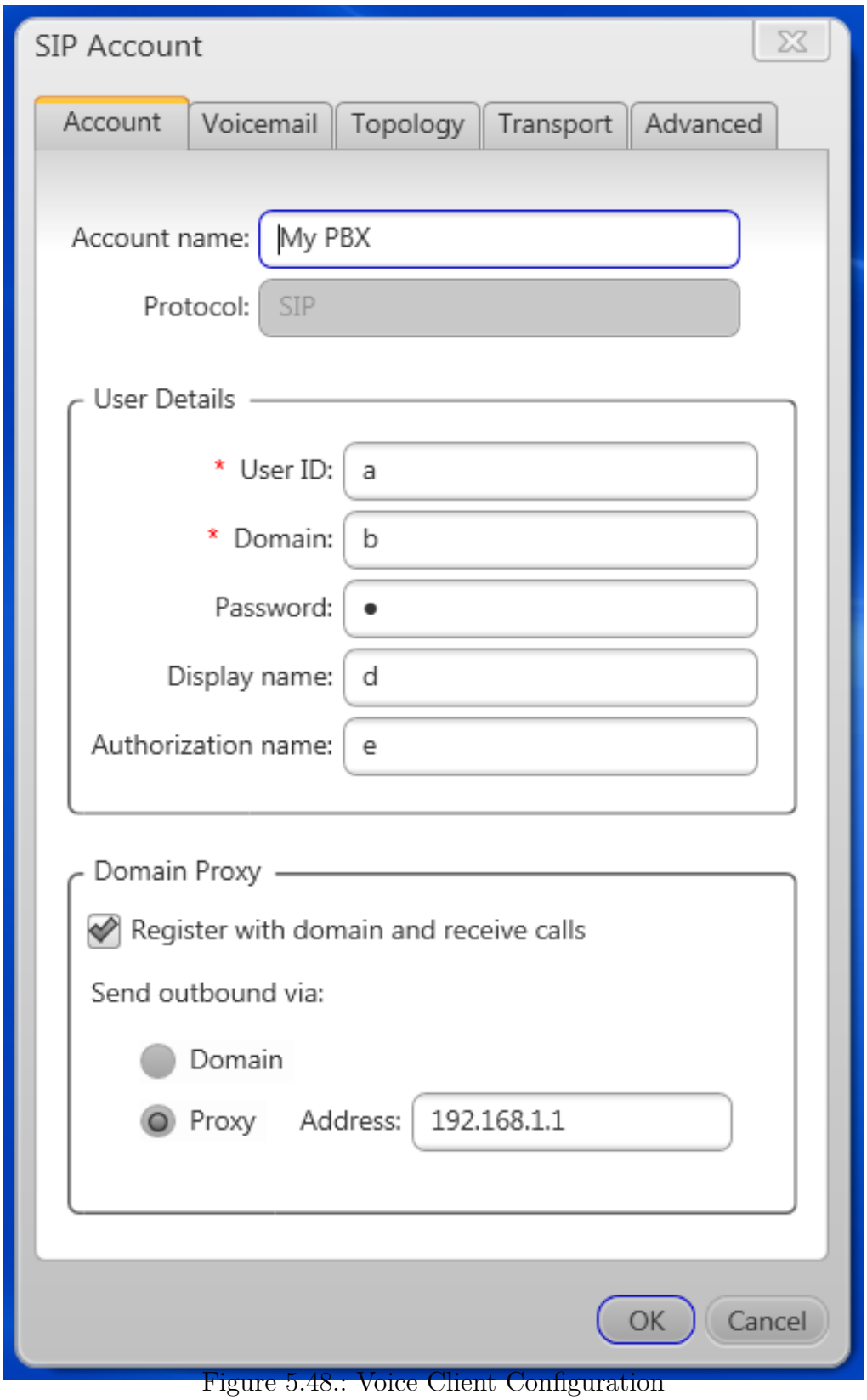


Figure 5.48.: Voice Client Configuration

5.8. SYSTEM

5.8.1. System

System Settings

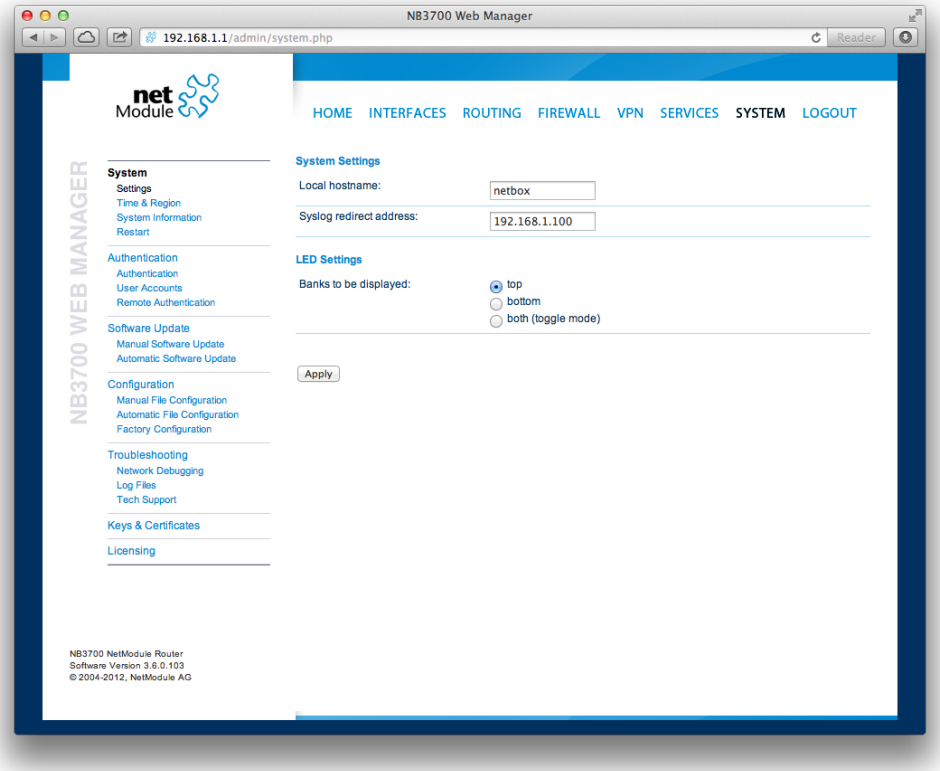


Figure 5.49.: System

The following system parameters can be set:

Local hostname: The hostname of the system

Syslog redirect address: Specifies an IP address to which system log messages should be redirected to. A tiny system log server for Windows is included in TFTP32 which can be downloaded from our website.

Syslog max. file size: The maximum size of message log files in kilobytes until they will be rotated

Reboot delay: The number of seconds which will be waited before regular system reboots (might be needed for `system-rebooting` events)

Banks to be displayed: You can configure the behavior of the status LEDs on the front panel of your device. They are usually divided into two banks (top/ bottom) and are either indicating the connection status or the digital IO port status. You may

configure toggle mode, so that the LEDs periodically cycle between the two states.

Time & Region

This page can be used for setting the system time and configuring the time zone. You may further enable daylight saving changes (e.g. automatically switching from summer to winter time) for your specific time zone.

NetModule routers can synchronize their system time by using one or more servers by the help of the Network Time Protocol (NTP) or via GPS. If enabled, the time synchronization is usually triggered after a WAN link has come up but before starting any VPN connections. Further time synchronization cycles are scheduled in background.

Time Synchronisation: Enable/disable time synchronization

NTP server: Address of the primary NTP server

NTP server 2: Optionally, the address of a second NTP server

Sync time from GPS: Derive time from first GPS device (if enabled)

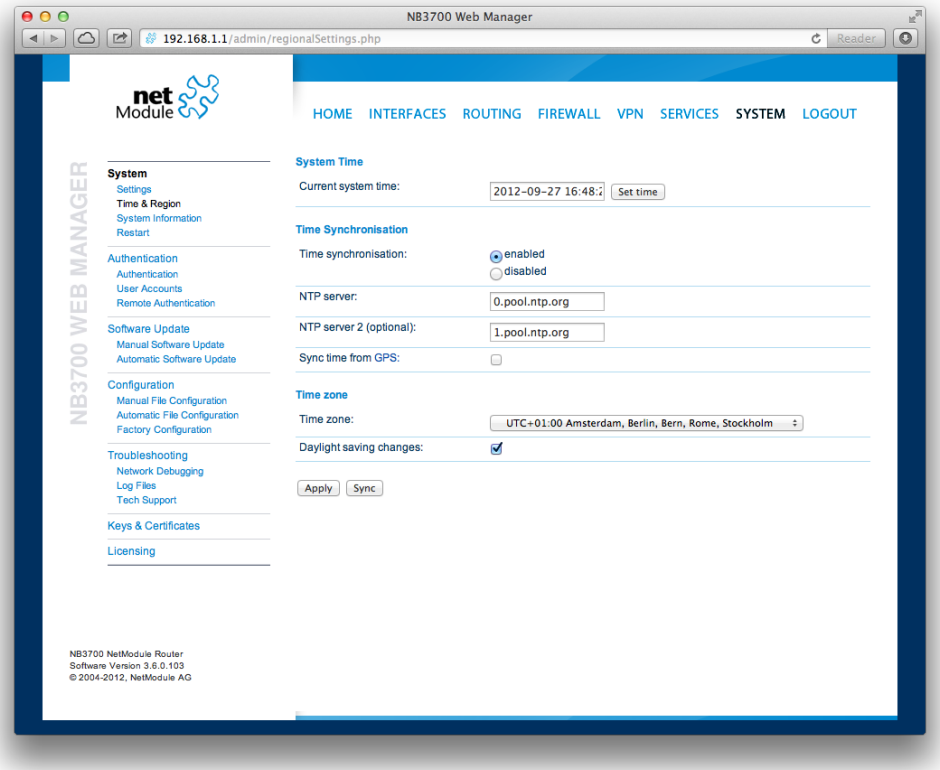


Figure 5.50.: Regional settings

System Information

The system information page displays various details of your NB3700, including system details, information about mounted modules and software release information.

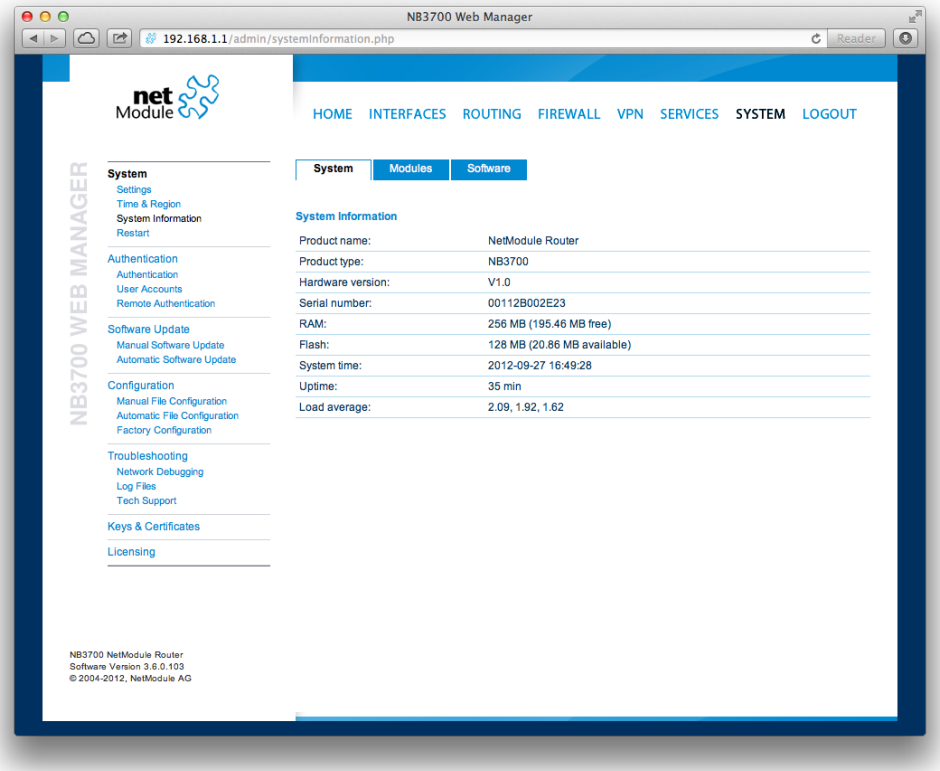


Figure 5.51.: System information

Restart

This link can be used to perform a restart of the system.

5.8.2. Authentication

This pages offers a simple shortcut to only allow secure connections (SSH, HTTPS) for managing the router.

User Accounts

By using this page you can manage the user accounts on the system. The standard `admin` user is a built-in power user that has permission to access the Web Manager and other administrative services and is used by several services as default user. Keep in mind that the `admin` password will be also applied to the `root` user which is able to enter a system shell.

Any other user represents a user with lower privileges, for instance it has only permission to view the status page or retrieve status values when using the CLI.

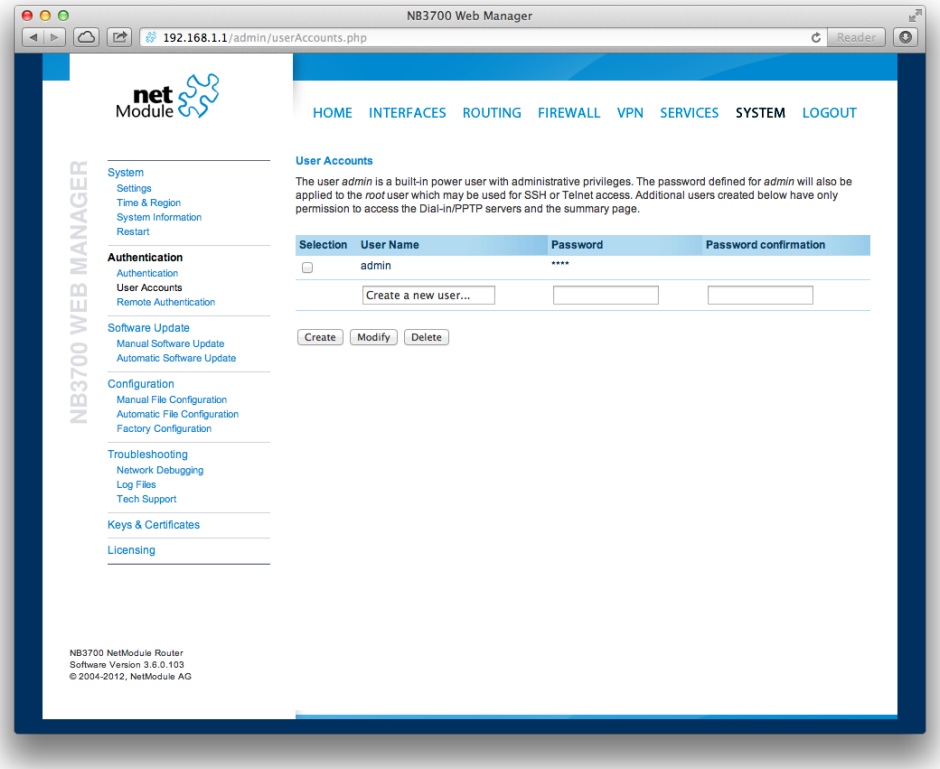


Figure 5.52.: User Accounts

User name: The name of the user (avoid whitespaces or special chars)

Password: The password of the user

Password confirmation: The confirmed password of the user

You will be able to modify or delete existing users here as well.

Remote Authentication

A RADIUS server can be used for authenticating remote users. This applies for the Web Manager, the WLAN network and other services supporting and incorporating remote authentication.

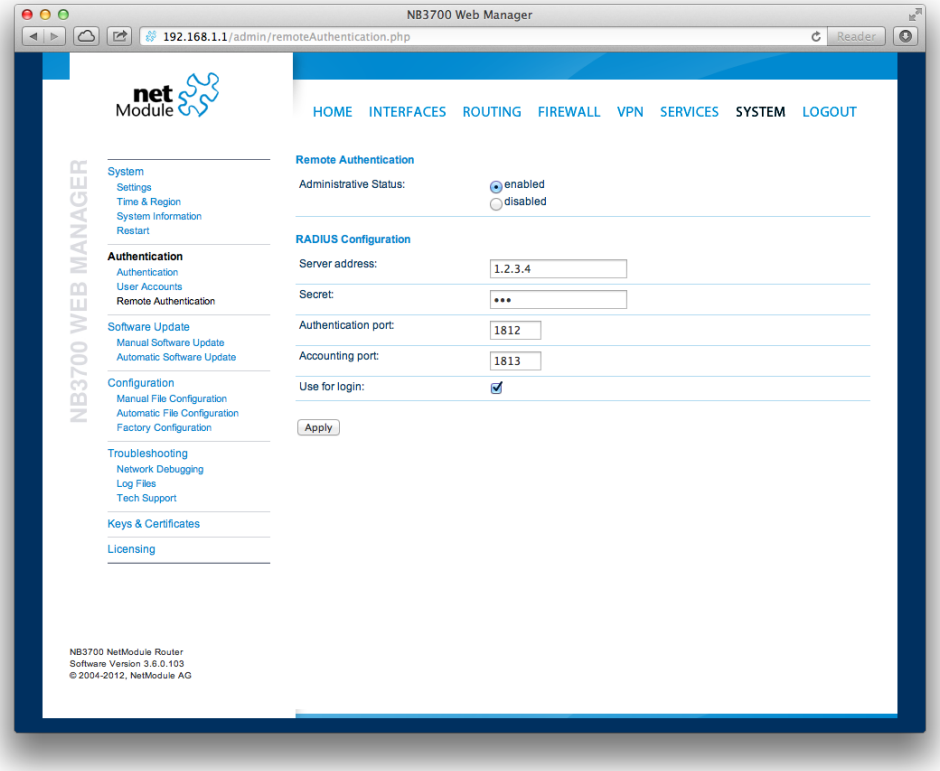


Figure 5.53.: Remote Authentication

It can be configured as follows:

Administrative status: Defines whether a remote server should be used for authentication

RADIUS server: The RADIUS server address

RADIUS secret: The secret used to authenticate against the RADIUS server

Authentication port: The port used for authentication

Accounting port: The port used for accounting messages

Use for login: This option enables remotely-defined users to access the Web Manager, otherwise it is only used by services which have explicitly configured it (e.g. WLAN)

5.8.3. Software Update

Manual Software Update

This menu can be used to run a manual software update of the system.

Update operation: The update operation method being used. You can upload the image, download it from an URL or use the latest version from our server

URL: The server URL where the software update image should be downloaded from. Supported protocols are TFTP, HTTP, HTTPS, and FTP. Provide a URL like `protocol://server/path/file`

Automatic Software Update

This menu can be used to run a automatic software update of the system.

Status: Enable/disable automatic software update

Time of day: Every day at this time the router will do a check for updates

URL: The server URL where the software update package should be downloaded from. Supported protocols are TFTP, HTTP, HTTPS, and FTP. Provide a URL like `protocol://server/path/file`

5.8.4. Configuration

Configuration via the Web Manager becomes tedious for larger volumes of devices. The router therefore offers automatic and manual file-based configuration to automate things. Once you have successfully set up the system you can back up the configuration and restore the system with it afterwards. You can either upload a single configuration file (.cfg) or a complete package (.zip) containing the configuration file and a packed version of other essential files (such as certificates) in the root directory.

Manual File Configuration

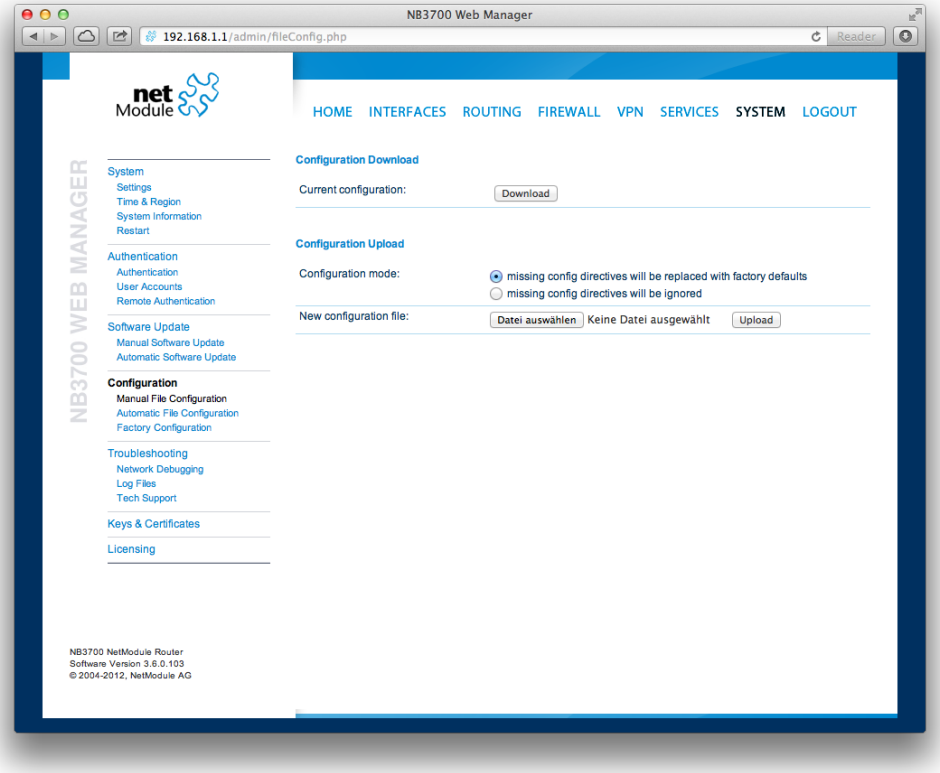


Figure 5.54.: Manual File Configuration

This section can be used to download the currently running system configuration (including essential files such as certificates). In order to restore a particular configuration you can upload a configuration previously downloaded. You can choose between missing configuration directives set to factory defaults or getting ignored, that means, potentially existing configuration directives will be kept at the system.

Automatic File Configuration

This menu can be used to run an automatic configuration update of the system. It is configured as follows:

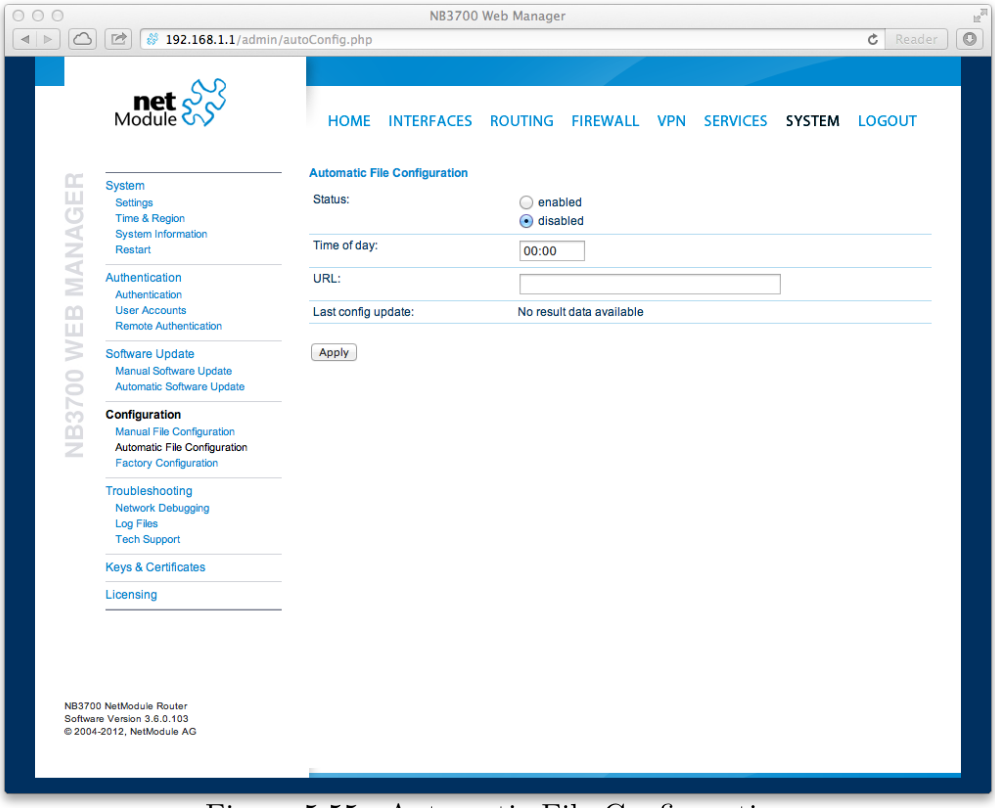


Figure 5.55.: Automatic File Configuration

Status: Enable/disable an automatic configuration update

Time of day: Time of day when the system should check for updates

URL: The URL where the configuration file should be retrieved from (supported protocols are HTTP, HTTPS, TFTP, FTP)

Factory Configuration

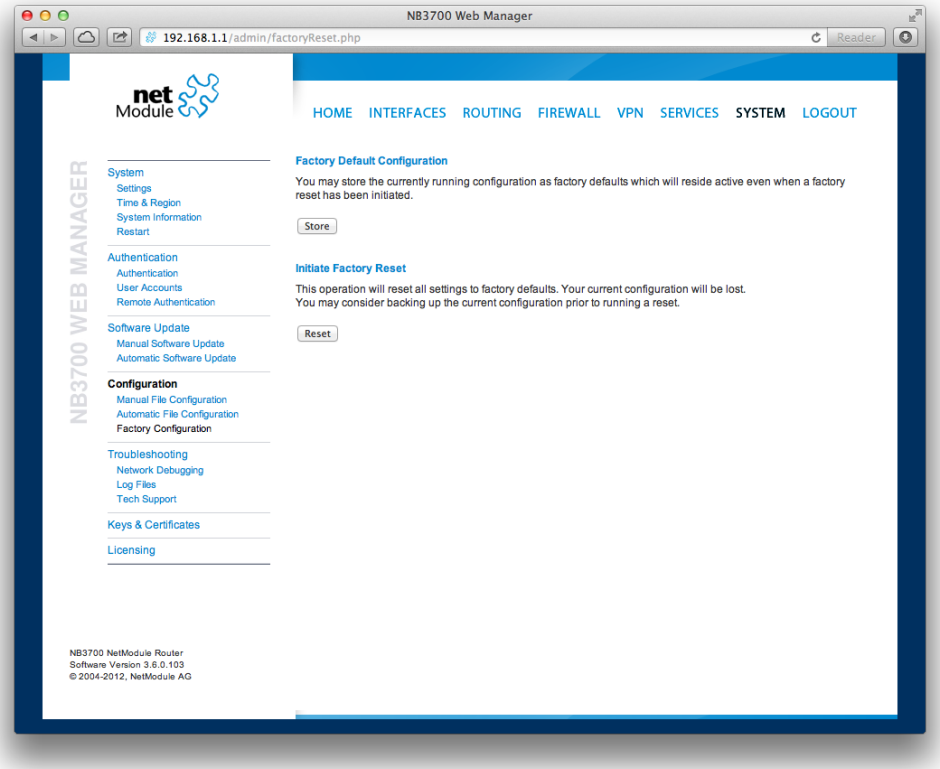


Figure 5.56.: Factory Configuration

This menu can be used to reset the device to factory defaults. Your current configuration will be lost. This procedure can also be initiated by pressing and holding the *Reset* button for at least five seconds. A successfully initiated factory reset can be noticed by all LEDs having been turned on. The factory reset will set the IP address of the first Ethernet interface back to 192.168.1.1. You will be able to communicate again with the device using the default network parameters. You may store the currently running configuration as factory defaults which will reside active even when a factory reset has been initiated (e.g. by your service staff).

Please ensure that this corresponds to a working configuration. A real factory reset to the default settings can be achieved by restoring the original factory configuration and initiating the factory reset again.

5.8.5. Troubleshooting

Network Debugging

Log Files

You can view the system log here by selection the option *Debug log* or if you are interested in the boot log select *Boot log*.

Another way to see what is going on on the box is opening a SSH or Telnet session as *root* and typing `tail-log`. Furthermore the system log can be redirected to a syslog server, see section 5.8.1.

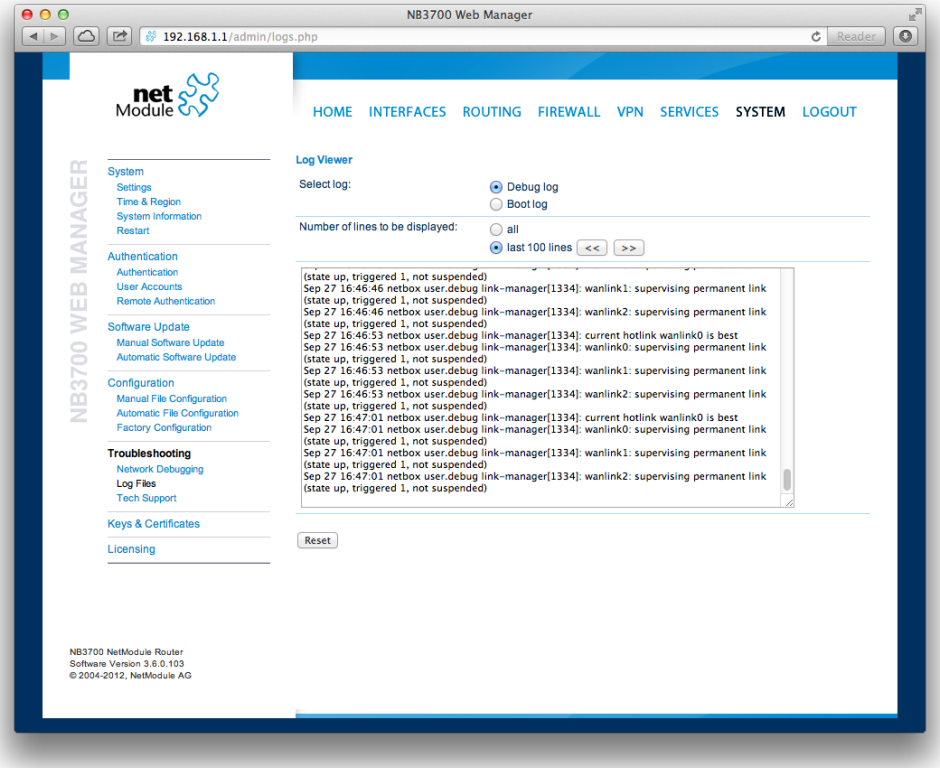


Figure 5.57.: Log Viewer

Tech Support

You can generate and download a tech support file here. We strongly recommend providing this file when getting in touch with our support team, either by e-mail or via our on-line support form, as it would significantly speed up the process of analyzing and resolving your problem. Log files can be viewed a downloaded and reset here. Please study them carefully in case of any issues. Various tools reside on this page for further analysis of potential configuration issues.

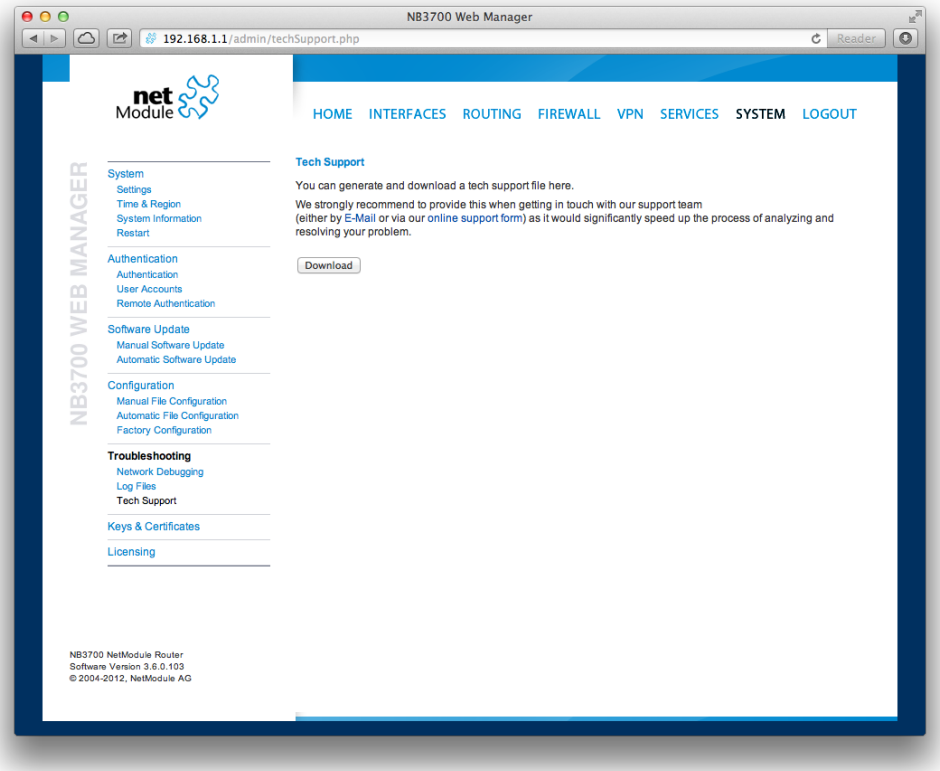


Figure 5.58.: Tech Support File

It is possible to trace any IP interface and inspect individual packet flows between hosts. This can be achieved by logging onto the box and start a network packet capture by using the tool *tcdump*. We recommend to use the -n switch to bypass name resolution (e.g. `tcpdump -n -i lan0`). You may also generate a dump in PCAP format using the Web Manager, download it to your computer and perform further inspections with Wireshark (available at www.wireshark.org).

5.8.6. Keys and Certificates

The key and certificate page lets you generate required files for securing your services (such as the HTTP and SSH server).

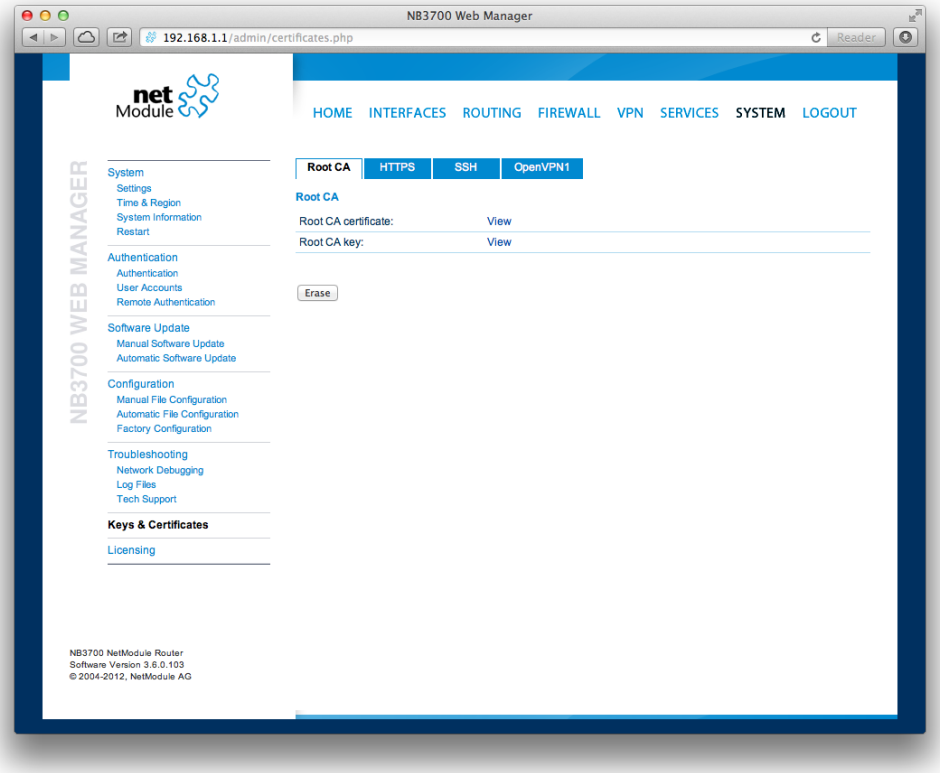


Figure 5.59.: Keys and certificates management

The following terms are used:

Term	Description
Root CA	The root Certificate Authority (CA) which issues certificates, its key can be used to certify it at trusted third party on other systems
Certificate	Corresponds to a digital certificate which uses a signature to bind a public key with an identity
Key	Corresponds to an either public or private key
CSR	Certificate Signing Request, which can be used to sign a certificate by a third party authority

Term	Description
P12	PKCS12 container format which can include certificates and keys protected by password
RSA	An encryption algorithm based on the fact that factorization of large integers is difficult
DSS/DSA	An encryption algorithm based on the discrete logarithm problem
Phrase	A password used for protecting keys

Table 5.7.: Certificate/Key Terms

A single certificate can obtain the following ASN.1 attributes:

Attribute	Description
CN	The certificate owner's common name, mainly used to identify a host
C	The certificate owner's country (usually a TLD abbreviation)
ST	The certificate owner's state
L	The certificate owner's location
C	The certificate owner's country
O	The certificate owner's organization
OU	The name of the organizational unit to which the certificate issuer belongs
E	The certificate owner's email address

Table 5.8.: Certificate Attributes

Those attributes form a so-called subject name, mainly used for matching a certificate or when signing certificate requests:

```
Subject: C=CH, ST=Switzerland, L=Zurich, O=Company, OU=Networking,
         CN=router.company.com/Email=info@company.com
```

Depending on your configuration, keys and certificates may be used for particular services, for instance if OpenVPN uses a certificate-based authentication or if you want to access the Web Manager over HTTPS.

Please note that an accurate system time is needed prior to creating certificates as it influences the lifetime of a certificate. The validity period is usually set to 10 years. You can further revoke and invalidate client certificates again (for instance if they have been compromised or lost).

5.8.7. Licensing

Certain features of NetModule routers require a valid license to be present in the system, some of them also depend on the mounted modules. Please contact us for getting a valid license for available components and we will provide a license file based on your serial number which can be installed to the router afterwards.

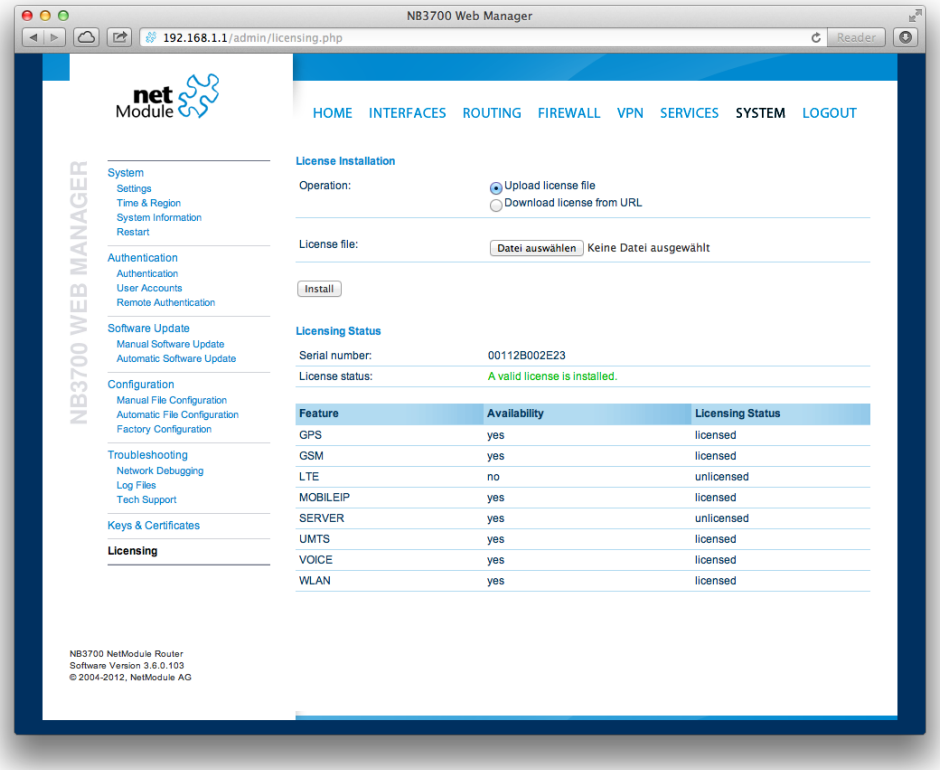


Figure 5.60.: Licensing

5.9. LOGOUT

Please use this menu to log out from Web Manager.

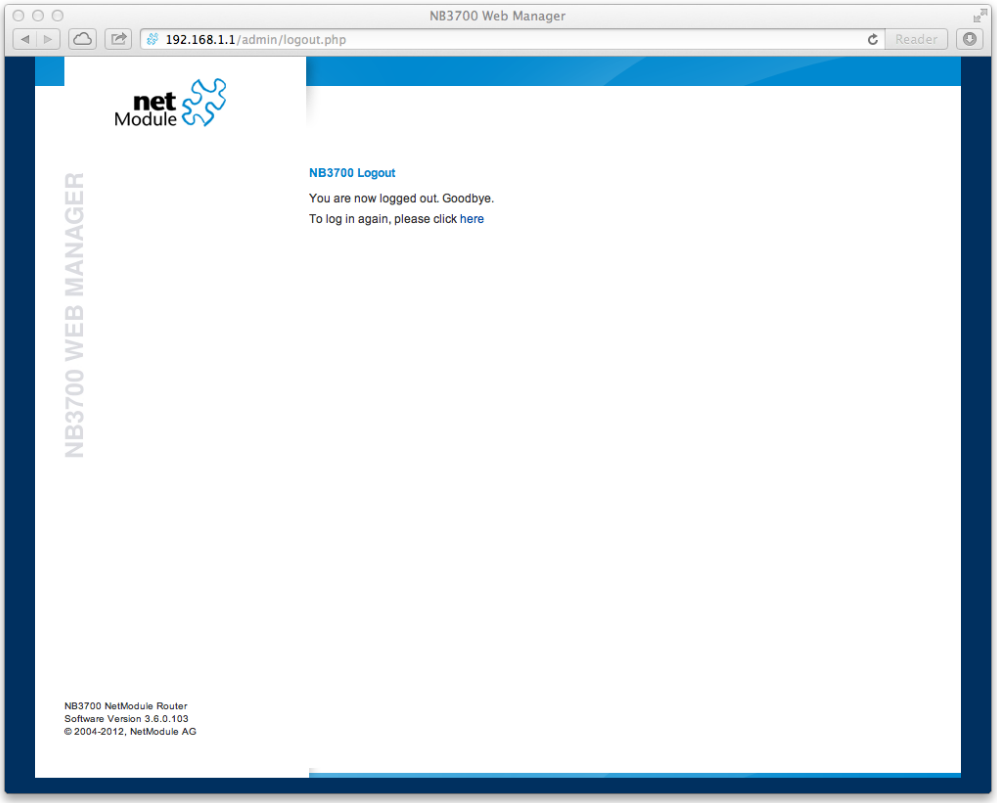


Figure 5.61.: Logout

6. Command Line Interface

ihe Command Line Interface (CLI) offers a unified control interface to the router and can be used to get/set configuration parameters, apply updates, restart services or perform other system tasks.

It will be started automatically in interactive mode when logging in as *admin* user or by running `cli -i`. However, the same syntax can be used when calling it from the system shell. A list of available commands can be displayed by running `cli -l`.

The CLI supports TAB completion, that is expanding entered words or fragments by hitting the TAB key at any time. This applies to commands but also to arguments and generally offers a convenient way for working on the shell.

Please note that each CLI session will perform an automatic logout as soon as a certain time of inactivity (10 minutes by default) have been reached. It can be turned off by the command `no-autologout`.

The CLI can be exited by running `exit`.

6.1. General Usage

When operating the CLI in interactive mode, each entered command will be executed by the `RETURN` key. You can use the `Left` and `Right` keys to move the current point between entered characters or use the `Up` and `Down` keys to search the history of entered commands. Pressing `CTRL-c` twice or `CTRL-d` on an empty command line will exit the CLI.

List of supported key sequences:

Key Sequence	Action
`CTRL-a`	Move to the start of the current line.
`CTRL-e`	Move to the end of the line.
`CTRL-f`	Move forward a character.
`CTRL-b`	Move back a character.
`ALT-f`	Move forward to the end of the next word.
`ALT-b`	Move back to the start of the current or previous word.

Key Sequence	Action
CTRL-l	Clear the screen leaving the current line at the top of the screen, with an argument given refresh the current line without clearing the screen.
CTRL-p	Fetch the previous command from the history list, moving back in the list.
CTRL-n	Fetch the next command from the history list, moving forward in the list.
ALT-<	Move to the first line in the history.
ALT->	Move to the end of the input history.
CTRL-r	Search backward starting at the current line and moving up through the history.
CTRL-s	Session will be frozen.
CTRL-q	Reactivate frozen session.
CTRL-d	Delete character at point or exit CLI if at the beginning of the line.
CTRL-t	Drag the character before point forward moving point forward as well. If point is at the end of the line, then this transposes the two characters before point.
ALT-t	Drag the word before point past the word after point, moving point over that word as well. If point is at the end of the line, this transposes the last two words on the line.
CTRL-k	Delete the text from point to the end of the line.
CTRL-y	Yank the top of the deleted text into the buffer at point.

Please note, that it can be required to apply quotes (") when entering commands with arguments containing whitespaces.
The following sections are trying to explain the available commands.

6.2. Print Help

The `help` command can be used to get the list of available commands when called without arguments, otherwise it will print the usage of the specified command.

```
> help
```

```
Usage:
        help [<command>]

Available commands:

        get                     Get config parameters
        set                     Set config parameters
        status                  Get status information
        send                    Send message or mail
        update                  Update system facilities
        restart                 Restart service
        reset                   Reset system to factory defaults
        reboot                  Reboot system
        shell                   Run shell command
        help                    Print help for command
        no-autologout           Turn off auto-logout
        exit                    Exit
```

6.3. Getting Config Parameters

The `get` command can be used to get configuration values.

```
> get -h
Usage:
        get [-hsvlc] <parameter> [<parameter>..]

Options:
        -s      generate sourceable output
        -v      validate config parameter
        -l      use legacy syntax with '&' separator
        -c      show configuration sections (can match a pattern)
```

6.4. Setting Config Parameters

The `set` command can be used to set configuration values.

```
> set -h
Usage:
        set [-hvl] <parameter>=<value> [<parameter>=<value>..]

Options:
```

```
        -v      validate config parameter
        -l      use legacy syntax with '&' separator
```

6.5. Getting Status Information

The `status` command can be used to get various status information of the system.

```
> status -h
Usage:
        status [-hs] <section>

Options:
        -s      generate sourceable output

Available sections:

        config                  Current configuration
        summary                 Short status summary
        system                  System information
        license                 License information
        wwan                    WWAN module status
        wlan                    WLAN module status
        gnss                    GNSS (GPS) module status
        lan                     LAN interface status
        wan                     WAN interface status
        openvpn                 OpenVPN connection status
        ipsec                   IPsec connection status
        pptp                    PPTP connection status
        dialin                  Dial-In connection status
        dio                     Digital IO status
        neigh                   Neighborhood status
        location                Current Location
```

6.6. Sending E-Mail or SMS

The `send` command can be used to send a message via E-Mail/SMS to the specified address or phone number.

```
> send -h
Name:
        cli-send (Send message or mail)
```

```
Usage:
        send [-h] <type> <dest> <msg>

Options:
        <type>      type of message to be sent (mail or sms)
        <dest>      destination of message (mail-address or phone-
        number)
        <msg>       message to be sent
```

6.7. Updating System Facilities

The `update` command can be used to perform various system updates.

```
> update -h
Usage:
        update [-hr] <software|config|license|sshkeys> <URL>

Options:
        -r      reboot after update

Available actions:

        software            Perform software update
        config              Update configuration
        license             Update licenses
        sshkeys             Install SSH authorized keys

You may run 'update software latest' to install the latest version.
```

6.8. Restarting Services

The `restart` command can be used to restart system services.

```
> restart -h
Usage:
        restart [-h] <service>

Available services:

        link-manager          WAN links
```

```
wwan-manager          WWAN manager
wlan                  WLAN interfaces
network               Networking
dnsmasq               DNS/DHCP server
configd               Configuration daemon
firewall              Firewall and NAPT
lighttpd              HTTP server
openvpn               OpenVPN connections
ipsec                 IPsec connections
pptp                  PPTP connections
snmpd                 SNMP daemon
syslog                Syslog daemon
telnet                Telnet server
dropbear              SSH server
vrrpd                 VRRP daemon
usbipd                USB/IP daemon
surveyor              Supervision daemon
voiced                Voice daemon
gpsd                  GPS daemon
smsd                  SMS daemon
```

6.9. Resetting System

The `reset` command can be used to reset the router back to factory defaults.

```
> reset -h
Usage:
        reset [-h]
```

6.10. Rebooting System

The `reboot` command can be used to reboot the router.

```
> reboot -h
Usage:
        reboot [-h]
```

6.11. Running Shell Commands

The `shell` command can be used to execute a system shell and run any arbitrary application.

```
> shell -h
Usage:
        shell [-h] [<cmd>]
```

6.12. CLI-PHP

CLI-PHP, an HTTP frontend to the CLI application, can be used to configure and control the router remotely. It is enabled in factory configuration, thus can be used for deployment purposes, but disabled as soon as the admin account has been set up.

The service can later be turned on/off by setting the `cliphp.status` configuration parameter:

```
    cliphp.status=0         Service is disabled
    cliphp.status=1         Service is enabled
```

This section describes the CLI-PHP interface for Version 2, the general usage is defined as follows:

```
Usage:
    http(s)://cli.php?<key1>=<value1>&<key2>=<value2>..<keyN>=<
    valueN>

  Available keys:

  output          Output format (html, plain)
  usr             Username to be used for authentication
  pwd             Password to be used for authentication
  command         Command to be executed
  arg0..arg31     Arguments passed to commands

Notes:
    The commands correspond to CLI commands as seen by 'cli -l', the
     arguments
    (arg0..arg31) will be directly passed to the cli application.

     Thus, an URL containing the following sequence:

        command=get&arg0=admin.password&arg1=admin.debug&arg2=admin.
        access

    will lead to cli being called as:
```

```
        $ cli get "admin.password" "admin.debug" "admin.access"

    It supports whitespaces but please be aware that any special
    characters in the URL
    must be specified according to RFC1738 (which usually done by
    common clients such
    as wget, lynx, curl).

Response:
    The returned response will always contain a status line in the
    format:

    <return>: <msg>

    with return values of OK if succeeded and ERROR if failed. Any
    output
    from the commands will be appended.

Examples:

    OK: status command successful
    ERROR: authentication failed
```

status - Display status information

```
Key usage:
        command=status[&arg0=<section>]
Notes:
        Available sections can be retrieved by running command=
        status&arg0=-h.
        System status can be displayed without authentication.

Examples:

http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=status&arg0=-h
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=status&arg0=summary
http://192.168.1.1/cli.php?version=2&output=html&command=status
```

get - Get configuration parameter

```
Key usage:
```

```
command=get&arg0=<config-key>[&arg1=<config-key>..]

Examples:


http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=get&arg0=config.version

http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=get&arg0=openvpn.status&arg1=snmp.status&arg2=ipsec.
status
```

set - Set configuration parameter

```
Key usage:
     command=set&arg0=<config-key>&arg1=<config-value>[&arg2=<config
     -key>&arg3=<config-value>..]
Notes:
    In contrast to the other commands, this command requires a set
    of tuples because of the reserved '=' char, i.e.
    [arg0=key0, arg1=val0], [arg2=key1, arg3=val1], [arg4=key2, arg5
    =val2], etc

Examples:



http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=set&arg0=snmp.status&arg1=1

http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=set&arg0=snmp.status&arg1=0&arg2=openvpn.status&arg3
=1
```

restart - Restart a system service

```
Key usage:
    command=restart&arg0=<service>
Notes:
    Available services can be retrieved by running 'command=restart&
    arg0=-h'
```

```
Examples:


http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=restart&arg0=-h

http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=restart&arg0=link-manager
```

reboot - Trigger system reboot

```
Key usage:
    command=reboot

Examples:


http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=reboot
```

reset - Run factory reset

```
Key usage:
    command=reset

Examples:


http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=reset
```

update - Update system facilities

```
Key usage:
  command=update&arg0=<facility>&arg1=<URL>
Notes:
    Available facilities can be retrieved by running 'command=update
    &arg0=-h'

Examples:

http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
```

```
admin01&command=update&arg0=software&arg1=tftp://192.168.1.254/
latest

http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=update&arg0=config&arg1=tftp://192.168.1.254/user-
config.zip

http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
admin01&command=update&arg0=license&arg1=http://192.168.1.254/xxx.
lic
```

7. Technical Support

NetModule's mission statement is to provide you with state of the art products, technologies and services for your embedded applications. This certainly includes a professional and friendly team of support engineers which will be pleased to offer consultancy, provide assistance and deliver solutions in case of technical issues. With their broad-based experience they will be able to narrow down your problem and thus prevent you from getting too much gray hair.

In case of support requests please use our support form and submit a detailed description of your problem together with a tech-support file which contains all the necessary information to speed up the process of analyzing and resolving your problem.

The latest software and documentation material can found in the technical support area via the NetModule website at `http://www.netmodule.com/products/support`.

Feedback

Your feedback is highly appreciated; please send comments, suggestions, feature requests, error reports or your personal user experience with this NB3700 router to router@support.netmodule.com.

8. Legal Notice

Copyright

This document contains proprietary information of NetModule. No parts of the work described herein may be reproduced. Reverse engineering of the hardware or software is prohibited and protected by patent law. This material or any portion of it may not be copied in any form or by any means, stored in a retrieval system, adopted or transmitted in any form or by any means (electronic, mechanical, photographic, graphic, optic or otherwise), or translated in any language or computer language without the prior written permission of NetModule.

The information in this document is subject to change without notice. We would like to point out that NetModule makes no representation or warranties with respect to the contents herein and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this information.

This document may contain information about third party products or processes. Such third party information is generally out of influence of NetModule and therefore NetModule shall not be responsible for the correctness or legitimacy of this information. If you experience any incorrect or erroneous specifications in the documentation, please report them in writing by email to router@support.netmodule.com. While due care has been taken to deliver accurate documentation, NetModule does not warrant that this document is error-free.

NetModule and *NB3700* are trademarks and the logo is a service mark of NetModule AG, Switzerland.

All other products or company names mentioned herein are used for identification purposes only and may be trademarks or registered trademarks of their respective owners. The following description of software, hardware or process of NetModule or other third party provider may be included with your product and will be subject to the software, hardware or other license agreements.

Contact

Please contact us for up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, press releases or any other concerns.

NetModule AG
Meriedweg 11
CH-3172 Niederwangen
Switzerland

Tel +41 31 985 25 10
Fax +41 31 985 25 11
info@netmodule.com
http://www.netmodule.com

Copyright ©2012 NetModule AG, Switzerland All rights reserved

A. Appendix

A.1. Abbrevations

Parameter	Description
ETHx	Corresponds to Ethernet interfaces (either single or switched ones)
LANx	LAN interfaces which are generally based on Ethernet interfaces (including bridges)
WLANx	Refers to a Wireless LAN interface which will be represented as additional LAN interface when configured as access point
WWANx	Refers to a Wireless Wide Area Network (2G/3G/4G) connection
TUNx	Specifies an OpenVPN tunnel interface (based on TUN)
TAPx	Specifies an OpenVPN tunnel interface (based on TAP)
PPTPx	Specifies a PPTP tunnel interface
MOBILEIPx	Refers to a Mobile IP tunnel interface
SIMx	Specifies the SIM slot as seen on the front panel
GNSSx	Specifies a Global Navigation Satellite System module
Mobilex	Identifies a WWAN modem
SERIALx	Identifies a serial port
OUTx	Specifies a digital I/O output port (DOx)
INx	Specifies a digital I/O input port (DIx)
ANY	Generally includes all options offered by the current section
APN	Access Point Name
CID	A Cell ID is a generally unique number used to identify each Base Transceiver Station (BTS).

Parameter	Description
LAC	The Location Area Code corresponds to an identifier of a set of base stations that are grouped together to optimize signaling
LAI	The Location Area Identity is a globally unique number that identifies the country, network provider and location area
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
DNS	Domain Name System
NAPT	Network Address and Port Translation
DHCP	Dynamic Host Configuration Protocol
SDK	Script Development Kit which can be used to program applications
CLI	Command Line Interface, a generic interface to query the router or perform system tasks
SIM	Subscriber Identity Module
SMS	Short Message Service
SSID	Service Set Identifiers, can be used to define multiple WLAN networks on a module
STP	Spanning Tree Protocol
USSD	Unstructured Supplementary Service Data
VRRP	Virtual Router Redundancy Protocol
VPN	Virtual Private Network
WAN	WAN links include all Wide Area Network interfaces which are currently activated in the system
FQDN	Fully qualified domain name

Table A.1.: Abbreviations

In general, internal interfaces are written lower-case and may have a different naming. Their index starts from zero, whereas interfaces seen by the user will be written in capital letters starting from one.

A.2. System Events

Event	Description
wan-up	WAN link came up
wan-down	WAN link went down
dio-in1-on	DIO IN1 turned on
dio-in2-on	DIO IN2 turned on
dio-in1-off	DIO IN1 turned off
dio-in2-off	DIO IN2 turned off
dio-out1-on	DIO OUT1 turned on
dio-out2-on	DIO OUT2 turned on
dio-out1-off	DIO OUT1 turned off
dio-out2-off	DIO OUT2 turned off
gps-up	GPS signal is available
gps-down	GPS signal is not available
openvpn-up	OpenVPN connection came up
openvpn-down	OpenVPN connection went down
ipsec-up	IPsec connection came up
ipsec-down	IPsec connection went down
pptp-up	PPTP connection came up
pptp-down	PPTP connection went down
dialin-up	Dial-In connection came up
dialin-down	Dial-In connection went down
mobileip-up	Mobile IP connection came up
mobileip-down	Mobile IP connection went down
system-login-failed	User login failed
system-login-succeeded	User login succeeded
system-logout	User logged out
system-rebooting	System reboot has been triggered

Event	Description
system-startup	System has been started
sdk-startup	SDK has been started
sms-sent	SMS has been sent
sms-received	SMS has been received
sms-report-received	SMS report has been received
call-incoming	A GSM call is coming in
call-outgoing	Outgoing GSM call is being established
ddns-update-succeeded	Dynamic DNS update succeeded
ddns-update-failed	Dynamic DNS update failed
usb-storage-added	USB storage device has been added
usb-storage-removed	USB storage device has been removed
system-time-updated	System time has been updated
test	test event

Table A.2.: System Events

A.3. Factory Configuration

The factory configuration including default values for any configuration parameter can be derived from the file `/etc/config/factory-config.cfg` on the router. You may also call `cli get -f <parameter>` for obtaining a specific default value.

A.4. SNMP VENDOR MIB

```
-- ************************************************************************
-- NB VENDOR MIB
--
--
-- (c) COPYRIGHT 2012 by NetModule AG, Switzerland
-- All rights reserved.
--
--
-- ************************************************************************

NB-MIB DEFINITIONS ::= BEGIN

-- ************************************************************************
-- imports
-- ************************************************************************

IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       Integer32, Counter32, Gauge32,
       Counter64, TimeTicks                    FROM SNMPv2-SMI
       TEXTUAL-CONVENTION, DisplayString,
       PhysAddress, TruthValue, RowStatus,
       TimeStamp, AutonomousType, TestAndIncr  FROM SNMPv2-TC
       MODULE-COMPLIANCE, OBJECT-GROUP         FROM SNMPv2-CONF
       snmpTraps                               FROM SNMPv2-MIB
       URLString                               FROM NETWORK-SERVICES-MIB
       enterprises                             FROM RFC1155-SMI;

-- ************************************************************************
-- module definition
-- ************************************************************************

nb MODULE-IDENTITY
        LAST-UPDATED "201211151000Z"
        ORGANIZATION "NetModule AG"
        CONTACT-INFO
            "NetModule AG, Switzerland"
        DESCRIPTION
            "MIB module which defines the NB router specific entities"
        REVISION "201211151000Z"
        DESCRIPTION
            "MIB for NB software releases >= 3.5"
       ::= { netmodule 10 }


-- ************************************************************************
-- root anchor
-- ************************************************************************

netmodule OBJECT IDENTIFIER ::= { enterprises 31496 }


-- ************************************************************************
-- table definitions
-- ************************************************************************

system        OBJECT IDENTIFIER ::= { nb   1 }
products      OBJECT IDENTIFIER ::= { nb  10 }
admin         OBJECT IDENTIFIER ::= { nb  40 }
wwan          OBJECT IDENTIFIER ::= { nb  50 }
gnss          OBJECT IDENTIFIER ::= { nb  51 }
dio           OBJECT IDENTIFIER ::= { nb  53 }
wlan          OBJECT IDENTIFIER ::= { nb  60 }
traps         OBJECT IDENTIFIER ::= { nb 100 }

-- ************************************************************************

nb1600        OBJECT IDENTIFIER ::= { products 46 }
nb2700        OBJECT IDENTIFIER ::= { products 47 }
nb3700        OBJECT IDENTIFIER ::= { products 48 }

-- ************************************************************************
-- NBAdminTable
-- ************************************************************************

swVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION
        "The currently installed system software version"
    ::= { admin 1 }

kernelVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The currently installed kernel version"
    ::= { admin 2 }

serialNumber OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The serial number of the device"
    ::= { admin 3 }

deviceRestart OBJECT-TYPE
    SYNTAX  INTEGER {
                restart (1)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Force a device restart"
    ::= { admin 10 }

configUpdate OBJECT-TYPE
    SYNTAX  URLString
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Update the system configuration from the specified URL.
          The URL must be preceded by one of the prefixes tftp://, ftp://, http://
         and point to a server directory which contains a file named <serial-number>.zip"
    ::= { admin 11 }

configUpdateStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                succeeded (1),
                failed (2),
                inprogress (3),
                notstarted (4)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The status of the last configuration update cycle"
    ::= { admin 12 }

softwareUpdate OBJECT-TYPE
    SYNTAX  URLString
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Update the system software from the specified URL,
          the URL must be preceded by one of the prefixes tftp://, ftp://, http://
         and point to a server directory which contains the relevant update files"
    ::= { admin 13 }

softwareUpdateStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                succeeded (1),
                failed (2),
                inprogress (3),
                notstarted (4)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The status of the last software update cycle"
    ::= { admin 14 }


-- ****************************************************************************
-- NBWwanTable
-- ****************************************************************************



nbWwanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NBWwanEntry
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The table describing all WWAN modems and their current settings"
    ::= { wwan 1 }

nbWwanEntry OBJECT-TYPE
    SYNTAX      NBWwanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry describing a WWAN modem and its current settings"
    INDEX       { wwanModemIndex }
    ::= { nbWwanTable 1 }


NBWwanEntry ::= SEQUENCE {
    wwanModemIndex Integer32,
    wwanModemName DisplayString,
    wwanModemType DisplayString,
    wwanServiceType DisplayString,
    wwanRegistrationState DisplayString,
    wwanSignalStrength Integer32,
    wwanNetworkName DisplayString,
    wwanLocalAreaIdentification DisplayString,
    wwanLocalAreaCode DisplayString,
    wwanCellId DisplayString
}

wwanModemIndex OBJECT-TYPE
    SYNTAX      Integer32(0..254)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "WWAN modem index"
    ::= { nbWwanEntry 1 }

wwanModemName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WWAN modem name"
    ::= { nbWwanEntry 2 }

wwanModemType OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WWAN modem type"
    ::= { nbWwanEntry 3 }

wwanServiceType OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current service type of the WWAN modem"
    ::= { nbWwanEntry 4 }

wwanRegistrationState OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current registration state of the WWAN modem"
    ::= { nbWwanEntry 5 }

wwanSignalStrength OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current signal strength of the WWAN modem (-999 equals unknown)"
    ::= { nbWwanEntry 6 }

wwanNetworkName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current Local Area Identification (LAI) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 7 }

wwanLocalAreaIdentification OBJECT-TYPE
```

```
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current LAI to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 8 }

wwanLocalAreaCode OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current Local Area Code (LAC) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 9 }

wwanCellId OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current Cell ID (CID) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 10 }

-- ****************************************************************************
-- NBGnssTable
-- ****************************************************************************

nbGnssTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NBGnssEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table describing all GNSS devices and their current settings"
    ::= { gnss }

nbGnssEntry OBJECT-TYPE
    SYNTAX      NBGnssEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry describing a GNSS device and its current settings"
    INDEX       { gnssIndex }
    ::= { nbGnssTable 1 }

NBGnssEntry ::= SEQUENCE {
    gnssIndex Integer32,
    gnssName DisplayString,
    gnssSystem DisplayString,
    gnssLat DisplayString,
    gnssLon DisplayString,
    gnssAlt DisplayString,
    gnssNumSat Integer32
}


gnssIndex OBJECT-TYPE

    SYNTAX      Integer32(0..254)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "GNSS device index"
    ::= { nbGnssEntry 1 }

gnssName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "GNSS device name"
    ::= { nbGnssEntry 2 }

gnssSystem OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "GNSS system used by the device"
    ::= { nbGnssEntry 3 }

gnssLat OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "The latitude value received by the GNSS device"
    ::= { nbGnssEntry 4 }

gnssLon OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The longitude value received by the GNSS device"
    ::= { nbGnssEntry 5 }

gnssAlt OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The altitude value received by the GNSS device"
    ::= { nbGnssEntry 6 }

gnssNumSat OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of available satellites for the GNSS device"
    ::= { nbGnssEntry 7 }

-- ********************************************************************************
-- NBDioTable
-- ********************************************************************************

dioStatusIn1 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Current value of digital I/O port IN1"
    ::= { dio 1 }

dioStatusIn2 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Current value of digital I/O port IN2"
    ::= { dio 2 }

dioStatusOut1 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Current value of digital I/O port OUT1"
    ::= { dio 3 }

dioStatusOut2 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Current value of digital I/O port OUT2"
    ::= { dio 4 }

dioSetOUT1 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Update value for digital I/O port OUT1"
    ::= { dio 10 }
```

```
dioSetOUT2 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Update value for digital I/O port OUT2"
    ::= { dio 11 }

-- ***************************************************************************
-- NBWlanTable
-- ***************************************************************************

nbWlanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NBWlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table describing all WLAN modems and their current settings."
    ::= { wlan }

nbWlanEntry OBJECT-TYPE
    SYNTAX      NBWlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry describing a WLAN modem and its current settings."
    INDEX       { wlanModuleIndex }
    ::= { nbWlanTable 1 }

NBWlanEntry ::= SEQUENCE {
    wlanModuleIndex Integer32,
    wlanModuleName DisplayString,
    wlanModuleType DisplayString,
    wlanNumClients Integer32

}

wlanModuleIndex OBJECT-TYPE
    SYNTAX      Integer32(0..254)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WLAN module index"
    ::= { nbWlanEntry 1 }

wlanModuleName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WLAN module name"
    ::= { nbWlanEntry 2 }

wlanModuleType OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WLAN module type"
    ::= { nbWlanEntry 3 }

wlanNumClients OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Current number of clients connected to the WLAN module (if operated as access point)"
    ::= { nbWlanEntry 4 }

-- ***************************************************************************
-- trap objects
-- ***************************************************************************

notifs          OBJECT IDENTIFIER ::= { traps 0 }

system-shutdown NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Indicates a system shutdown"
    ::= { notifs 10 }

vpn-down-trap NOTIFICATION-TYPE
```

```
    STATUS current
    DESCRIPTION "Indicates a VPN connection deactivation"
    ::= { notifs 20 }

vpn-up-trap NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Indicates a VPN connection activation"
    ::= { notifs 21 }

wwan-rssi-low NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "The WWAN RSSI is below the specified threshold"
    ::= { notifs 30 }

END
```

A.5. SDK Examples

Event	Description
config-summary.are	This script shows a summary of the currently running configuration.
dio-monitor.are	This script monitors the DIO ports and sends a SMS to the specified phone number.
dio-server.are	This script implements a TCP server which can be used to control the DIO ports.
dio.are	This script can be used to set a digital output port.
email-to-sms.are	This script implements a lightweight SMTP server which is able to receive mail and forward them as SMS to a phone number.
gps-monitor.are	A script for activating WLAN as soon as GPS position (lat,lon) is within a specified range.
led.are	This script can be used to set a LED
mount-media.are	This script can be used to mount an USB storage stick.
read-config.are	This script can be used to read a configuration parameter.
send-mail.are	This script will send an E-Mail to the specified address.
send-sms.are	This script will send an SMS to the specified phone number.
serial-read.are	This script can be used to write a message to the serial port.
serial-readwrite.are	This script will write and read to the serial port.
serial-tcsetattr.are	This script can be used to set/get the attributes of the serial port.
serial-write.are	This script can be used to write a message to the serial port.
sms-control.are	This script will execute commands received by SMS.
sms-delete-inbox.are	This script can be used to flush the SMS inbox.
sms-read-inbox.are	This script can be used to read the SMS inbox.
sms-to-email.are	This script will forward incoming SMS messages to a given E-mail address.
sms-to-serial.are	This script can be used to write a received SMS to the serial port.

Event	Description
status.are	This script can be used to display all status variables
syslog.are	Throw a simple syslog message.
tcpclient.are	This script sends a message to a TCP server.
tcpserver.are	This script implements a TCP server which is able to receive messages.
udp-msg-server.are	This script will run an UDP server which is able to receive messages and forward them as SMS/E-Mail.
udpclient.are	This script sends a message to a remote UDP server.
udpserver.are	This script implements an UDP server which is able to receive messages.
write-config.are	This script can be used to set a configuration parameter.

Table A.3.: SDK Examples

A.6. SDK Sample SMS Control

```
/* This script will execute commands received by SMS and may report the status of the system. */

INTERVAL        = 10;                           /* only run every 10 seconds                    */
MAXMSG          = 5;                            /* process max. 5 msgs                          */
MAXAGE          = 300;                          /* message mustn't be older than 5 mins         */
MAXLINES        = 32;                           /* max. number of lines in msg (incl. header)   */
AUTH            = 1;                            /* perform authentication                       */
ADMPWD          = "";                           /* password used for authentication             */
SENDERS         = mkarray(/* "+123456789" */);  /* allowed senders                              */

/* ------------------------------ startup ---------------------------------------------------- */

/* check if we should perform authentication */
if (argc == 2 && argv[1] == "noauth") {
    AUTH = 0;
}

/* retrieve password */
if (AUTH) {
    if (strlen(ADMPWD) == 0) {
        /* use configured admin password */
        ADMPWD = nb_config_get("admin.password");
        if (strlen(ADMPWD) > 0) {
            nb_syslog("using admin password for authentication");
        } else {
            /* not there -> use a default password instead */
            nb_syslog("using default password for authentication");
            ADMPWD = "admin01";
        }
    }
}

/* ------------------------------ functions -------------------------------------------------- */

/* parse message */
string parse (string msg)
{
    /* read by line */
    lnr = 0;
    ishdr = 1;
    tlnr = 0;
    allowed = (length(SENDERS) == 0) ? 1 : 0;

    lp = msg;
    for (lnr = 0; lnr < MAXLINES && strlen(lp) > 0; lnr++) {
        pos = strchr(lp, "\n");
      if (is_void(pos)) pos = strlen(lp);

        line = left(lp, pos);
      lp = substr(lp, pos + 1);

        if (strlen(line) == 0) {
            /* saw header separator */
            ishdr = 0;
            continue;
        }
        if (ishdr) {
            /* saw header line */
            if (left(line, 5) == "Sent:") {
                /* check age of message */
                sentdate = trim(substr(line, 5));
                sent = strptime(sentdate, "%Y-%m-%d %H:%M:%S");

                if (!is_void(sent)) {
                    /* got a valid sent date */
                    now = localtime(time());
                    age = mktime(now) - mktime(sent);
                    nb_syslog("message has been sent %ds ago", age);
                    if (age > MAXAGE) {
                      nb_syslog("rejecting too old message");
                        return "";
                    }
                } else {
                    nb_syslog("time check has been omitted");
                }
            } else if (left(line, 5) == "From:") {
                from = substr(line, 6);
                if (length(SENDERS) > 0) {
                    for (s = 0; s < length(SENDERS); s++) {
                        sender = SENDERS[s];
```

```
                        if (left(from, strlen(sender)) == sender) {
                            allowed = 1;
                            break;
                        }
                    }
                }
                if (allowed == 0) {
                    nb_syslog("rejecting message from unknown sender %s", from);
                    return "";
                } else {
                    nb_syslog("sender %s can pass", from);
                }
            }
        } else {
            /* saw text line */
            if (AUTH && tlnr == 0) {
                /* first line of message must contain the password */
                if (left(line, strlen(ADMPWD)) != ADMPWD) {
                    nb_syslog("authentication failed");
                    return "";
                } else {
                    nb_syslog("authentication succeeded");
                }
            } else if ((AUTH && tlnr == 1) || (!AUTH && tlnr == 0)) {
                /* this line must contain the command */
                if (left(line, 6) == "reboot") {
                    return "reboot";
                } else if (left(line, 7) == "connect") {
                    return "connect";
                } else if (left(line, 10) == "disconnect") {
                    return "disconnect";
                } else if (left(line, 6) == "status") {
                    return "status";
                } else if (left(line, 6) == "output") {
                    printf("parsed msg: %s\n", left(line, 6));
                    return left(line, 13);
                }
            } else {
                break;
            }
            tlnr++;
        }
    }

    nb_syslog("no command detected");

    return "";
}

int setdio (string cmd)
{
    newstate = substr(cmd, 9, 3);
    port = substr(cmd, 7, 1);

    if (port != "1" && port != "2") {
        nb_syslog("invalid DIO port %s\n", port);
        return -1;
    }
    if (newstate == "on") {
        st = 1;
    } else if (newstate == "off") {
        st = 0;
    } else  {
        nb_syslog("invalid new DIO state %s\n", newstate);
        return -1;
    }

    rc = nb_dio_set(sprintf("out%s", port), st);
    if (rc) {
        nb_syslog("Unable to set state %s for DIO output port %s\n", newstate, dout);
        return -1;
    } else {
        nb_syslog("Setting state %s for DIO output port %s\n", newstate, dout);
        return 0;
    }
}



/* ------------------------------- main ------------------------------------------------------ */

/* check if we got stressed */
LASTFILE = "/tmp/sms-control.last";
last = 0;
```

```
fp = fopen(LASTFILE, "r");
if (fp) {
    /* we have been run at least one time */
    str = fread(fp, 32);
    if (str) last = (int) str;
    fclose(fp);
}

now = mktime(localtime(time()));
elapsed = now - last;
if (elapsed > 0 && elapsed < INTERVAL) {
    nb_syslog("we have ben run %d seconds ago. skipping.", elapsed);
    exit(0);
}

/* record timestamp */
fp = fopen(LASTFILE, "w+");
if (fp) {
    str = sprintf("%d", now);
    fwrite(fp, str);
    fclose(fp);
}

/* read inbox */
msgs = nb_sms_list();
nr_msgs = length(msgs);

if (nr_msgs == 0) {
    nb_syslog("there are no messages in your inbox");
    exit(0);
}

nb_syslog("you have %d message(s) in your inbox", nr_msgs);

/* track states */
reboot = 0;
connecting = 0;
disconnecting = 0;


/* only process latest messages */
start = nr_msgs - MAXMSG;
if (start < 0) start = 0;

for (i=start; i<nr_msgs; i++) {
    msg = nb_sms_retrieve(msgs[i]);
    if (!msg) continue;

    nb_syslog("processing message %d of %d (ID %s)",
              i, nr_msgs, msgs[i]);

    cmd = parse(msg);
    if (cmd == "reboot") {
        nb_syslog("reboot command received");
        reboot = 1;
    } else if (cmd == "connect") {
        nb_syslog("connect command received");
        if (connecting) {
            nb_syslog("already connecting");
        } else {
            /* enable first wanlink */
            nb_config_set("wanlink.0.mode=1");
            connecting = 1;
        }
    } else if (cmd == "disconnect") {
        nb_syslog("disconnect command received");
        if (disconnecting) {
            nb_syslog("already disconnecting");
        } else {
            /* disable first wanlink */
            nb_config_set("wanlink.0.mode=0");
            disconnecting = 1;
        }
    } else if (cmd == "status") {
        nb_syslog("status command received");
        rcpt = nb_sms_header(msgs[i], "From");
        if (rcpt) {
            id = nb_sms_send(rcpt, nb_status_summary());
            if (!id) {
                nb_syslog("unable to send status message to %s", rcpt);
            } else {
                nb_syslog("successfully queued status message to %s (ID %s)", rcpt, id);
            }
        }
```

```
    } else if (left(cmd,6) == "output") {
        nb_syslog("dio out command received");
        setdio(cmd);
    } else {
        nb_syslog("ignoring invalid message");
    }

    /* delete message  */
    ret = nb_sms_delete(msgs[i]);
    if (ret == 0) nb_syslog("deleted message %s", msgs[i]);
}

if (reboot == 1) {
    /* trigger reboot  */
    nb_syslog("rebooting system");
    nb_reboot();
}

exit(0);

/* -------------------------------- end of script ------------------------------------------------- */
```